

TMAP7 User Manual

G. R. Longhurst

October 2004



*Idaho National Engineering and Environmental Laboratory
Bechtel BWXT Idaho, LLC*

TMAP7 Manual

Glen R. Longhurst

Published October 2004

**Idaho National Engineering and Environmental Laboratory
Idaho Falls, Idaho 83415-3860**

**Prepared for the
U.S. Department of Energy
Office of Fusion Energy Sciences
Under DOE Idaho Operations Office
Contract DE-AC07-99ID13727**

ABSTRACT

The TMAP Code was written at the Idaho National Engineering and Environmental Laboratory by Brad Merrill and James Jones in the late 1980s as a tool for safety analysis of systems involving tritium. Since then it has been upgraded to TMAP4 and has been used in numerous applications including experiments supporting fusion safety, predictions for advanced systems such as the International Thermonuclear Experimental Reactor (ITER), and estimates involving tritium production technologies. Its further upgrade to TMAP2000 and now to TMAP7 was accomplished in response to several needs. TMAP and TMAP4 had the capacity to deal with only a single trap for diffusing gaseous species in solid structures. TMAP7 includes up to three separate traps and up to 10 diffusing species. The original code had difficulty dealing with heteronuclear molecule formation such as HD and DT under solution-law dependent diffusion boundary conditions. That difficulty has been overcome. TMAP7 automatically generates heteronuclear molecular partial pressures when solubilities and partial pressures of the homonuclear molecular species are provided for law-dependent diffusion boundary conditions. A further sophistication is the addition of non-diffusing surface species. Atoms such as oxygen or nitrogen or formation and decay or combination of hydroxyl radicals on metal surfaces are sometimes important in reactions with diffusing hydrogen isotopes but do not themselves diffuse appreciably in the material. TMAP7 will accommodate up to 30 such surface species, allowing the user to specify relationships between those surface concentrations and partial pressures of gaseous species above the surfaces or to form them dynamically by combining diffusion species or other surface species. Additionally, TMAP7 allows the user to include a surface binding energy and an adsorption barrier energy. The code includes asymmetrical diffusion between the surface sites and regular diffusion sites in the bulk. All of the previously existing features for heat transfer, flows between enclosures, and chemical reactions within the enclosures have been retained, but the allowed problem size and complexity have been increased to take advantage of the greater memory and speed available on modern computers. One feature unique to TMAP7 is the addition of radioactive decay for both trapped and mobile species. Whereas earlier versions required a separate FORTRAN compiler to operate, TMAP7 is based on a public-license compiler, distributed with the code.

FOREWORD

This report is intended to document the theoretical and practical bases for upgrading in the TMAP4 code (*TMAP4 User's Manual*, EGG-FSP-10315, June 12, 1992) to convert it to TMAP7. The model was assembled using simplified approximations to complex physical processes. The new version will add flexibility and some new capabilities in analyzing transport of dissolved gases through structures and within enclosures surrounding those structures. The previous versions of the code were written for use within the capabilities of second generation desk-top computers such as the X286 class machines. These had typically only 640 kilobytes of base memory available and operated with computational rates far below what is available now. TMAP7 assumes several tens of megabytes of RAM will be available, and its added complexity benefits from the higher operating speeds available on modern desk-top and work station computers. Further refinements and enhancements are hoped for in the future. The latter part of this report documents the specifics for code implementation including the details of the input file structure required to define problems for solution. It also provides some sample calculations to demonstrate some of the code's capabilities. Further sample problems are provided in the companion report, Ref. 1.

CONTENTS

ABSTRACT	iii
FOREWORD	iv
1.0 INTRODUCTION	1
2.0 CODE MODIFICATIONS	1
2.1 Compiler.....	1
2.2 Problem Size	1
3.0 MODEL DEVELOPMENT	4
3.1 Movement Across Surfaces	4
3.1.1 Dissociation/Recombination	4
3.1.2 Solution Law.....	8
3.1.3 Fixed Boundary	13
3.1.4 Nonflow	13
3.2 Movement from Surface to Bulk	14
3.3 Movement Through Structures.....	15
3.3 Structural Thermal Response.....	17
3.3.1 Surface Boundary Conditions	19
3.3.2 Composite Structure Treatment	19
3.4 Enclosure Response.....	20
4.0 INPUT FILE	25
4.1 Title Input.....	25
4.2 Main Input.....	25
4.2 Enclosure Input	28
4.2.1 Functional Enclosures.....	29
4.2.2 Boundary Enclosures	33
4.3 Thermal Input	35
4.4 Diffusion Input	40
4.5 Equation Input	51
4.6 Table Input	52
4.7 Control Input	53
4.8 Plot Input.....	55
4.9 Terminal Declaration.....	57
5.0 CODE USAGE	59
5.1 Installation.....	59
5.1.1 General	59
5.1.2 Setup.....	61
5.2 Operation.....	61
5.2.1 TMAP7 Code	61
5.2.2 Extract Utility	62
5.3 Restart Capability	63
6.0 CONCLUSION	65
ACKNOWLEDGEMENT	65
REFERENCES.....	66
APPENDIX BATCH FILE LISTINGS	67

1.0 INTRODUCTION

The first version of TMAP was made available by the Idaho National Engineering and Environmental Laboratory (INEEL) Fusion Safety Program in 1988. Known as TMAP/MOD1, it had limited distribution.² That code was upgraded, verified and validated^{3,4} as the TMAP4 code in 1992. Since then, there have been advances in machine capability and operating systems and improvements in understanding of the processes involved in solute atom transport. The INEEL Fusion Safety Program, who developed and sponsored this code, determined that it would be appropriate to upgrade the TMAP4 code. An intermediate version, TMAP5, was never formally distributed. The next version was TMAP2000, which was developed but never verified and validated. The addition of radioactive decay was the sixth revision to make TMAP7. Because TMAP4 is the version in general use, discussion here will emphasize changes since that version.

Several objectives were in mind during the revision of TMAP4 to make TMAP7, the most recent version with significantly greater capability. These were:

- The ability to deal with multiple traps allowing different capture rates for trapping different species and different release rates from the various traps in the various diffusion segments.
- An adjusted physical model for heteronuclear^a molecule recombination at surfaces under solution-law dependent diffusion boundary conditions.
- Built-in capacity to deal with radioactive decay of selected species.
- Extended limits on numbers of nodes, segments, etc.
- Increased problem memory allowances to take advantage of the capabilities of modern computers.
- Elimination of the need to purchase a FORTRAN compiler to use the TMAP7 program.

^a Heteronuclear molecules are composed of different atoms or isotopes such as HD, while homonuclear molecules have identical atoms, for example H₂.

2.0 CODE MODIFICATIONS

Here we address major changes implemented to bring the code to its present status.

2.1 Compiler

A first task was to consider a compiler that has no or much relaxed inherent limitations regarding the size of the executable program it can produce in comparison with the previous commercially available compiler used. One public-license compiler available is the GNU FORTRAN 77 compiler. While not as up-to-date as some of the commercial compilers, it compiles programs executable from various Windows® environments, or more specifically, from the DOS window in these operating systems. It also works well on UNIX platforms. There is also a graphics package available to use with it, and it is freely exportable for non-commercial purposes.

2.2 Problem Size

As previously configured, the maximum size limitations in TMAP4 are as stated in the TMAP4 user manual (see Ref. 3).

TMAP7 allows up to 50 separate diffusion segments (thermsegs/diffsegs), each of which can be considered a parallel flow path with the others or linked together to form composite structures. The total number of nodes allowed is 1000. For memory allocation reasons, there is a limit of 100 on the product of number of nodes in any one linked structure and the number of traps active in that structure. Each thermseg/diffseg end can still interact with only one enclosure per face.

The linking process for combining segments into paths remains the same as in prior versions. However, in the past there was a requirement for the segments to be linked in numerical order. That restriction has been removed, although each segment may still only be linked to at most one other segment at each face.

The maximum number of 10 diffusing species could probably have been reduced, though for permeable membranes it was considered useful to keep it at 10. Most applications to date have been limited to three (e.g., H, D, and T). The need has sometimes arisen to consider an additional category of species for surface nodes: chemisorbed atoms or complexes that do not diffuse through the medium. The number of such species has been set at 30 in TMAP7. These are in addition to the 10 diffusing species. Those surface species are considered as a separate species class with a diffusivity or surface mobility but only present at the surface. In addition, there are instances where surface binding energies must be considered separately from ordinary dissociation and recombination. Hence, the option has been added to indicate when such treatment is necessary and identify surface binding energies and chemisorption activation energies for all surface species and diffusion species when this option is active.

Thermal calculations in individual thermsegs/diffsegs remain optional. Such calculations may include conduction, convection, and heat generation.

Trapping calculations in individual thermsegs/diffsegs also remain optional. However, the trapping process has been modified to include multiple traps and the trapping and release of multiple species in traps. In TMAP7, each *diffseg* may include up to 3 different trap types, and the user identifies the species competing for those traps. Furthermore, the ability to have spatially distributed trap concentrations has been added. Trapping rates are determined by the mobilities of the various species in diffusion and the trap concentrations. Trap release rates are thermally activated and thus are the same for all species in a given trap, depending on the local temperature.

The maximum number of defined enclosures remains at 40. Each enclosure may still interact with more than one thermseg/diffseg but not vice-versa. For boundary type enclosures, we have added the ability to have pre-programmed enclosure temperatures as well as pressures defined as functions of time by equations or tables. The temperature specification as a function of time is also now available in functional enclosures. Further, the code will determine heteronuclear molecular gas pressures when homonuclear pressures are specified for species that combine to form them under the functional enclosure rules for law-dependent diffusion boundary conditions. This determination is not needed for boundary enclosures.

The maximum number of 100 defined flow paths for convective flow between enclosures was considered to be adequate. That gives each of the 40 possible enclosures at least one exit and still leaves 60 for cross-linking.

The maximum number of defined enclosure species was increased to 30. This is because some of the allowed 10 diffusion species may react to form homonuclear (e.g., $2\text{H} \rightarrow \text{H}_2$) and heteronuclear ($\text{H}+\text{D} \rightarrow \text{HD}$) molecules (6 for hydrogen isotopes alone). There should also be allowances for oxygen, nitrogen, and hydrocarbon reactants and reaction products such as water. Enclosure species counterparts are needed for exchange with surface-only species as well.

The maximum number of chemical reactions that may be defined for operation in enclosures continues to be adequate at 50.

In reality, there is no present limit on the number of user-defined equations. The two-step process is retained of (1) processing the input file and compiling the EQU subroutine as TAPE7.O, then (2) linking TAPE7.O with TMAPC7.O and performing the execution, taking input and state parameters from the TAPE1 file, though all that is accomplished automatically by the available .BAT files.

The maximum number of tables has been increased to 100 with an aggregate of 550 data pairs.

The TMAPINP file is organized in the same general sequence as with TMAP4 for functionality and simple economy of effort. Most applications are made by modifying an existing file, and editing is easier if one knows his or her way around to start with. However, there are differences in structure within the various blocks to accommodate new features.

No change was required in the policy that allows individual entries within the data blocks to be order-independent. The major data blocks, themselves, will continue to require inclusion in the specified order.

When iterative convergence is required, the convergence variables continues to be the empty trap concentrations when trapping is active or surface species concentrations. However, Gaussian iteration has replaced Jacobian matrix inversion as the solution algorithm to accommodate surface species interactions.

3.0 MODEL DEVELOPMENT

This section documents much of the theoretical development for TMAP7. In some instances, this merely repeats what was presented in Ref. 3. Differences from TMAP4 are emphasized.

3.1 Movement Across Surfaces

Gas movement across solid surfaces is often governed by molecular dissociation to solid solution and the reverse process of molecular recombination to the gas phase at the surface of the solid. For many situations with metals, this process is sufficiently close to equilibrium that a solution law such as Sieverts' law applies. For non-metals, solution usually follows Henry's law, and it is molecules that are absorbed and released. Adsorption and release of molecules from a surface not in equilibrium with the surrounding gas may be rate-limited by chemisorption which is characterized by an Arrhenius expression as are other thermally activated processes. TMAP7 has the ability to simulate these processes.

3.1.1 Dissociation/Recombination

Two kinds of dissociation/recombination are considered in TMAP7. Those are conventional, referred to heretofore as *ratedep*, and a new *surfdep* or recombination and release limited by defined surface energies. Conventional *ratedep* recombination formulates the generation and release rate of molecules as the product of two atom concentrations at the surface and a recombination rate coefficient, often in Arrhenius form, that relates molecular production rate to the concentrations of the constituent atoms or complexes at the surface. Molecules formed under the *ratedep* specification are assumed to come off readily. For *surfdep* recombination, the production rate to form surface species proceeds as the product of random lateral jumps, but release is thermally activated and involves the surface binding energy explicitly. Further, when this option is active, the difference in surface binding energy and solution enthalpy is considered in establishing surface concentrations of diffusion species.

When conventional (*ratedep*) dissociation and recombination govern, there is some dispute about whether it is only surface atoms that take part in the recombination process or whether atoms within a lattice constant or two of the surface will also take part.⁵ TMAP7 assumes the former.

Under *ratedep* conditions, molecules arriving at the surface from the gas above the surface are assumed to immediately dissociate and transform to two surface dissociation product atoms or complexes. Under *surfdep* conditions, only adsorption takes place from the gas. Subsequent processes of dissociation, combination with other species, or decay are considered independently.

When surface binding energies are employed (*surfdep* boundary), surface concentrations of diffusion species are asymmetrically diffusively coupled (described later) with the

interior concentrations. Combinations of various species are also assumed to occur as the result of lateral diffusive movement over the surface under *surfdep* boundary conditions.

Only two-body recombination is considered in TMAP7. Therefore, for a polyatomic molecule such as water to form at the surface from oxygen and hydrogen, there first must be the formation of an OH at the surface, and it combines in turn with a diffusing hydrogen atom in a second reaction. In that case, the OH could be a surface species with a high binding energy. This is only calculated mechanistically under the *surfdep* and *ratedep* boundary conditions. For *lawdep* boundary conditions, rates of such reactions are based on the relative concentrations of the species that can participate in the combination and follow chemical equilibrium requirements. Reactions such as water formation can also be accounted for through enclosure chemical reactions for all boundary conditions.

We now consider these processes in more detail.

3.1.1.1 *Ratedep Boundary*

When conventional (*ratedep*) dissociation and recombination governs, the net flow of atoms of species “s” into the surface is given by

$$J_s = \sum_{m=1}^k a_{m_s} J_m \quad (1)$$

$$J_m = K_{d_m} P_m - \sum_{i,j} K_{r_m} C_i C_j \quad (2)$$

where:

J_s = atom flux of atomic species “s” into the surface from the enclosure

a_{m_s} = number of atoms of species “s” in a molecule of species “m”

J_m = molecular flux of molecular species “m” into the surface

K_{d_m} = dissociation coefficient of molecular species “m” at the surface

P_m = partial pressure of molecular species “m” above the surface

K_{r_m} = recombination coefficient for molecular species “m” which consists of one each of atomic species “i” and “j” (for homonuclear molecule s = j)

C_i = surface concentration of atomic or complex species “s”

C_j = surface concentration of conjugate atomic or complex species “j”

Note in Equation (2) that the summation is over all reactions that form molecular species "m" and counts species "i" combining with species "j" as well as the reverse. Surface-only species are assumed to be in volumetric solution, but only in the surface monolayer of lattice atoms.

Mass conservation at the surface requires that for species "s" at the surface

$$\begin{aligned} \sum_i a_{m_i} (P_{m_i} K_{d_{m_i}} - K_{r_{m_i}} C_s C_j) + D_s \left(\nabla C_s + C_s \frac{Q_s^*}{kT^2} \nabla T \right) \\ + \sum_j \frac{C_s C_j D_s}{N\lambda} + C_s \lambda \left[\nu_o \exp\left(-\frac{E_b}{kT}\right) - \nu_s \right] = 0 \end{aligned} \quad (3)$$

Here

C_s = concentration of species "s" at the surface

D_s = diffusivity of species "s" near the surface

Q_s^* = heat of transport or Ludwig-Soret coefficient for atomic species "s" in the material

ν_o = Debye frequency ($\sim 10^{13} \text{ s}^{-1}$)

N = lattice atom density (atom/m³)

λ = lattice parameter (m)

k = Boltzmann's constant

T = temperature (K)

E_b = binding energy for molecule dissociation

ν_s = radioactive decay frequency for atoms of species "s"

The net flux to the surface of molecular species "m" is given by the first term in parentheses, where the sum is over all molecules that contain the species s. The second term in parentheses is the atom flux diffusing from the bulk to the surface. The third term (second sum) is the rate at which species "s" particles jump to adjacent sites and combine with a conjugate species, C_j , to form a different species. The summation is over all such combinations that are defined to be operating involving species "s" particles. The last term is the rate at which species "s" particles dissociate into constituents. The binding energy, E_b , for the dissociation of the molecule, if any, because it is in the solid state, will in general be different from the formation free energy in the gas or liquid phases. Because this is a purely surface species concentration, a point value, there is no inertial term.

3.1.1.2 *Surfdep*

If recombination is surface-energy limited (*surfdep*), transport of atoms to and from the surface is described by similar equations, but we consider the formation and release of a molecule as a two-step process. The formation rate of a molecular surface species is reflective of Equation (3).

$$\begin{aligned} \frac{dC_m}{dt} = & \sum_i \frac{C_i C_j}{N} \left(\frac{D_i + D_j}{\lambda^2} \right) - C_m \sum_j \frac{C_j}{N} \left(\frac{D_i + D_j}{\lambda^2} \right) - C_m \nu_o \exp\left(-\frac{E_b}{kT}\right) \\ & + \sum_l C_l \nu_o \exp\left(-\frac{E_{b_l}}{kT}\right) \end{aligned} \quad (4)$$

Here, the first term is the formation of the surface species from two constituents. The summation is over all species “i” that can combine with another species “j” to produce the “m” molecule. For heteronuclear molecule formation, there are two identical terms in the sum as “i” atoms jump to meet “j” atoms and vice versa. The second sum is loss by combination with a second species to form a third. The third sum is loss by dissociation to form other species. The fourth is production by the dissociation of one or more other species. ν_o is again the Debye frequency which is associated with all thermally activated processes. The diffusivities shown are for lateral diffusion activation and should differ from the lattice diffusivity. It is assumed that energy liberated or consumed in the formation or dissociation of these particles is exchanged with the lattice, which is a sufficiently large sink that it has no significant contribution to the local temperature.

Molecular flux to and from the surface is then given by

$$J_m = \frac{P_m}{\sqrt{2\pi M k T}} \exp\left(-\frac{E_x}{kT}\right) - \frac{C_m \nu_o}{6} \exp\left(-\frac{E_x - E_c}{kT}\right) \quad (5)$$

where in addition to the previous definitions:

M = molecule mass

E_x = barrier energy for molecular entry to the surface, assumed positive only

E_c = surface binding energy of species m

The first term in Equation (5) is the molecule arrival rate coupled with a Boltzmann factor to give the frequency with which molecules overcome the surface barrier potential and attach to the surface. The second term is the frequency of molecule escape from the surface. It is assumed that chemical energy required or released by the molecule capture or liberation will be given up or absorbed by the lattice. That heat is considered negligible for the heat transfer part of the problem. The factor of 6 in the denominator of the last term is a steric factor to account for the probability that the direction of motion on any given jump will be

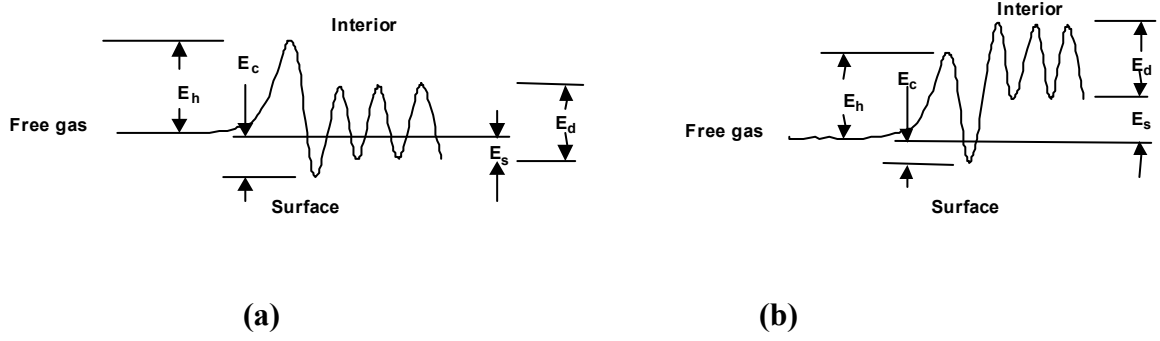


Figure 1. Comparison of (a) exothermic, and (b) endothermic solubility potentials, each with a surface binding energy, E_c , and a capture threshold energy, E_x .

toward the surface and thus result in escape from the surface. It implicitly implies a cubic structure. The energies involved are illustrated in Figure 1.

For surface-only species, the surface concentration must be such that transport from the surface by molecular fluxes is balanced by the net formation rate by combination reactions. In equation form

$$J_m + \frac{\partial C_{m_{form}}}{\partial t} = 0 \quad (6)$$

For diffusion species, mass conservation at the surface is given by

$$\delta x \frac{\partial C_m}{\partial t} = J_m + \delta x \frac{\partial C_{m_{form}} - \nu_m C_m}{\partial t} + D_m \left(\nabla C_m + C_m \frac{Q_m^*}{RT^2} \nabla T \right) \quad (7)$$

This is similar to Equation (6) except an inertial term, a diffusion term, and radioactive decay have been added.

3.1.2 Solution Law

Where movement governed by dissociation and recombination proceeds at a rate fast enough that there is effectively equilibrium between gaseous and solid phases of the solute, Sieverts' law applies, relating the equilibrium concentration of solute atoms in the metal to the gas partial pressure above the surface of the metal. This is the *lawdep* boundary condition. In that case, holdup at the surface due to a surface binding energy or finite dissociation or recombination rates will not be significant to the equilibrium. Furthermore, chemical equilibrium must be maintained among the various species in the enclosures where a *lawdep* diffusion boundary applies.

3.1.2.1 Sieverts Law

When molecules dissociate and recombine at the surface in equilibrium, Sieverts' law governs. The general form of the solution relation is

$$C_i C_j = K_{ij} P_m \quad (8)$$

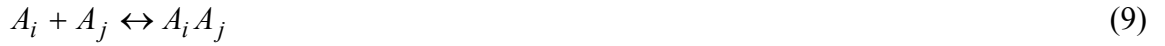
where:

C_i, C_j = concentrations at the surface of combining species

K_{ij} = equilibrium constant for molecular species “m” in the gas phase with its constituents, atom species “i” and “j” dissolved in the matrix.

P_m = partial pressure of molecular gas “m” above the surface

Note that this is merely a restatement of the law of mass action for the reaction



where the molecular species is made up of 1 atom or complex of species “i” and 1 atom or complex of species “j”. The equilibrium constant is related to the free energy of formation for the molecule, ΔG_f .

$$K_{ij} = \exp\left(-\frac{\Delta G_f}{kT}\right) \quad (10)$$

For homonuclear molecules, Equation (4) reduces to the more conventional form where the Sieverts' solubility becomes the square root of the equilibrium constant.

$$K_s = \exp\left(\frac{\Delta S_f}{2k}\right) \exp\left(-\frac{\Delta H_f}{2kT}\right) \equiv K_{so} \exp\left(-\frac{E_s}{kT}\right). \quad (11)$$

Here ΔS_f and ΔH_f are the formation entropy and enthalpy, respectively, for the dissolved atoms.

An implication of Equation (8) is that there is a requirement for both constituent atom or complex populations to be present at the surface if there is going to be a partial pressure P_m of the heteronuclear gas above the surface. Further, as a consequence of thermodynamics for surface reactions that involve heteronuclear molecules, it will be impossible to arbitrarily specify all the molecular gas partial pressures under a law-dependent (equilibrium) boundary condition in a *functional*-type enclosure. For *boundary*-type enclosures, which are quite artificial, the presence of the heteronuclear species is of no consequence and so is not computed.

To demonstrate the species interdependence, consider the situation where only atomic species A and B are at the surface, but they recombine to form three molecular species, A₂, AB, and B₂. If the system is in equilibrium, Sieverts' law will need to be satisfied for the two homonuclear molecules A₂ and B₂:

$$C_A = K_{s_{A_2}} \sqrt{P_{A_2}} \quad (12)$$

$$C_B = K_{s_{B_2}} \sqrt{P_{B_2}} \quad (13)$$

where the K_s terms are Sieverts' solubilities as defined in Equation (11). In true equilibrium, there is no net flow of either A or B atoms to the surface via the homonuclear molecule channels. There will also need to be no net flow of atoms to or from the surface for either of the species comprising the heteronuclear molecule AB:

$$J_A = J_B = P_{AB} K_{d_{AB}} - C_A C_B K_{r_{AB}} = 0. \quad (14)$$

Thus, the pressure P_{AB} will be fixed when C_A and C_B are fixed, i.e., when P_{A_2} and P_{B_2} are set:

$$P_{AB} = \frac{K_{r_{AB}}}{K_{d_{AB}}} C_A C_B = \frac{K_{r_{AB}}}{K_{d_{AB}}} K_{s_{A_2}} K_{s_{B_2}} \sqrt{P_{A_2} P_{B_2}}. \quad (15)$$

We now consider how $K_{r_{AB}}$ and $K_{d_{AB}}$ are related. Microscopically, recombination can be thought of as the jumping of an A atom or complex to a site where it finds a B atom or complex. Such jumping is thermally activated and is the product of a lattice vibrational or Debye frequency and a Boltzmann factor, $\exp\left(-\frac{E}{kT}\right)$, where E is the activation energy or the potential that the A atom must overcome to move from its initial site. Typically, that is the diffusion activation energy, E_d . That jump frequency is multiplied by the probability that the A atom or complex will find an A or B partner when it arrives at the new lattice site. That probability is essentially the concentration of the A or B species normalized by the lattice density. Thus, for homonuclear molecule, A₂, formation, the rate of production is proportional to the concentration of jumping atoms, C_A , and to the concentration of receiving atoms, also C_A , or to C_A^2 .

$$J_{r_A} = 2K_r C_A^2 \quad (16)$$

The initial factor of 2 on the right side of Equation (16) appears because 2 atoms of species A are included in each molecule of A₂. For dissociation of a homonuclear molecule, the arrival rate is taken to be the same factor of 2 multiplied by the kinetic molecular arrival rate,

$\frac{P_{A_2}}{\sqrt{2\pi M_{A_2} kT}}$ multiplied by a sticking factor, α , and a Boltzmann factor if the atoms must

overcome a potential peak, E_x , to gain access to the material. Hence, the dissociation rate may be expressed as

$$J_{d_A} = 2K_d P_{A_2} = 2 \frac{\alpha}{\sqrt{2\pi M_{A_2} kT}} \exp\left(-\frac{E_x}{kT}\right) P_{A_2}. \quad (17)$$

Figure 1 shows both exothermic and endothermic solution potential distributions schematically for cases where there is a surface binding energy different from the solution enthalpy. Positive potentials lie above the free gas line and negative ones below. In Figure 1(a), E_x is identical with E_h . In other cases, such as that shown in Figure 1(b), E_x will be simply the sum of the solution enthalpy per atom, E_s , and the diffusion activation energy, E_d , if that sum is greater than zero, but E_x has a minimum of zero.

Because under Sieverts' equilibrium the net flux of atoms to the surface must be zero and

$$C = K_s \sqrt{P} \quad (18)$$

Equations (16) and (17) may be combined to get (neglecting subscripts to identify species)

$$K_r = \frac{\alpha}{K_{s_o}^2 \sqrt{2\pi M kT}} \exp\left(\frac{2E_s - E_x}{kT}\right) \quad (19)$$

which is the Baskes form of the recombination rate coefficient for homonuclear molecules.⁶

Now for heteronuclear molecules, the arrival rate in dissociation will be similar, except that only one A atom or complex will arrive with each molecule and there may be a difference in the solution and diffusion activation energies for the two species. Their masses will also be different. Therefore, E_x may be more complex. For heteronuclear molecule formation, A atoms jump to meet B atoms, but B atoms also jump to meet A atoms. Thus the rate of production must be proportional to $2C_A C_B$, though only one A atom or complex is returned with each AB molecule. If we remove the initial factor of two in Equation (16) to recognize that only one A atom or complex comes off with the molecule, and substitute $2C_A C_B$ for C_A^2 , and combine the result with a similarly modified version of Equation (17), we get

$$K_{r_{AB}} = \frac{\alpha}{K_{s_{A_2}} K_{s_{B_2}} \sqrt{2\pi M_{AB} kT}} \exp\left(\frac{E_{s_i} + E_{s_j} - E_x}{kT}\right) \frac{P_{AB}}{2\sqrt{P_{A_2} P_{B_2}}} \quad (20)$$

Substituting these relationships back into Equation (15), assuming equivalent solubilities, means that

$$P_{AB} = 2\sqrt{P_{A_2} P_{B_2}} \quad (21)$$

We have implicitly assumed here, from the equivalent solubilities condition, that we are dealing with different isotopes of the same element. In that case, the same result could be obtained using standard thermodynamic arguments for an equilibrium chemical reaction involving the three species and the assumption that the configurational entropy for the heterogeneous molecule is twice that for the homogeneous molecule. Therefore, Equation (21) is the relationship to be used in a Sieverts' law dependent boundary condition for heteronuclear species formed of isotopes of the same element. A more general equation, applicable to all binary combinations, and the one used to establish heteronuclear molecular gas partial pressures in boundary-type enclosures in TMAP7, is

$$P_C = \eta P_A^{\nu_A} P_B^{\nu_B} \quad (22)$$

Here, η is a numerical coefficient, which may be a function of temperature, and the ν_i are stoichiometric coefficients.

When heteronuclear molecules are anticipated, only homonuclear molecular gas pressures can be specified or else one of the homonuclear molecule pressures must be left free while the other is specified. TMAP7 assumes the user has specified homonuclear pressures and automatically calculates heteronuclear pressures under equilibrium or *lawdep* conditions for *functional* type enclosures. For functional enclosures under *lawdep* conditions, slight departures from equilibrium are allowed such that there may be net surface flows to and from the enclosures defined by diffusive fluxes of diffusion species and specified reactions involving surface-only species.

Suppose that internal concentration gradient conditions result in diffusion fluxes J_A and J_B of species A and B to or from the surface. From species conservation, it must be true that

$$\begin{aligned} J_A &= J_{AB} + \frac{J_{A_m}}{\nu_A} \\ J_B &= J_{AB} + \frac{J_{B_n}}{\nu_B} \end{aligned} \quad (23)$$

where it is assumed that the equivalent chemical reaction is



and it is evident that

$$\begin{aligned} m &= \frac{1}{\nu_A} \\ n &= \frac{1}{\nu_B} \end{aligned} \quad (25)$$

In addition to the diffusive fluxes into or out from the enclosure, there may also be convective flows and generation or consumption rates for each or any of the species in the enclosure. For the species A_m , B_n , and AB involved in the *lawdep* equilibrium, there will be any necessary species-to-species conversions to ensure that

$$P_{AB} = \eta P_{A_m}^{\nu_A} P_{B_n}^{\nu_B} \quad (26)$$

while assuring conservation of the various A and B constituents involved.

3.1.2.2 Henry's Law

To this point, emphasis has been on dissociation-recombination processes at surfaces. When molecules enter and leave the surface without dissociation and recombination, and the process is in equilibrium, then Henry's law applies. Then, the concentration of molecules on the surface is related to the pressure of the gas above the surface with the constant of proportionality referred to as the Henry's law solubility.

$$C_m = K_h P_m \quad (27)$$

where

C_m = concentration of molecules of species "m" at the surface

K_h = Henry's law solubility coefficient.

Normally, Henry's law solubility is an Arrhenius expression similar to the one for Sieverts' solubility in Equation (11).

3.1.3 Fixed Boundary

For the *sconc* boundary condition, the concentration of each of the species, both diffusion and surface-only, is set by some defining function of time. It may be a constant or a value given by an equation or looked up with interpolation in a table. Because the fixed surface concentration boundary condition is artificial, it is meaningless to consider molecular flows to or from a *functional* enclosures, and none are calculated. Further, one should not invoke the *sconc* diffusive boundary condition for a diffusion structure adjacent to a *functional* enclosure.

3.1.4 Nonflow

In this boundary condition, there are no flows through surfaces. This is accommodated by forcing the concentration gradient to zero for diffusing species. It has no meaning for surface-only species.

3.2 Movement from Surface to Bulk

When there is a defined surface binding energy (*surfdep* conditions), an atom, molecule, or complex resident at the surface must overcome the binding potential to escape. In Figure 1, that potential is shown as the difference between E_h and E_c . Normally it will also mean that movement of diffusion species from the bulk to the surface takes place more readily than from the surface to the bulk. We refer to that here as *asymmetric diffusion*. Both processes are thermally activated, so we could describe the flux of atoms from a surface site to a bulk site by

$$J_b = \frac{\nu_o \lambda}{6} \left[C_o \exp\left(-\frac{E_s + E_d - E_c}{kT}\right) - C_I \exp\left(-\frac{E_d}{kT}\right) \right] \quad (28)$$

Here

C_o = concentration at surface site

C_I = concentration at subsurface site

The factor of 6 is a steric factor accounting for the supposition that atoms in each location have one chance in six of jumping in the direction of the other site given that on any particular attempt they exceed the potential threshold for that jump. Temperature differences are meaningless over distances of one lattice constant, so temperature gradient effects have been ignored. If there is thermodynamic equilibrium between the two sites, then

$$C_o = C_I \exp\left(\frac{E_s - E_c}{kT}\right) \quad (29)$$

We can relate Equation (28) to more familiar properties by considering the special case where $E_c = E_d$ and $E_h = E_s + E_d$, but not less than 0. That corresponds to simple diffusion through the bulk. Comparing Equation (28) with the discretized form of simple Fick's law diffusion flux between matrix layers, we see that $\frac{\nu_o \lambda}{6} = \frac{D_{so}}{\lambda}$ such that Equation (28) can be rewritten as

$$J_b = \frac{D_0}{\lambda} \left[C_0 \exp\left(\frac{E_c - E_s}{kT}\right) - C_I \right]. \quad (30)$$

Here the D_0 refers to the diffusivity at the surface, not to the pre-exponential coefficient in the expression for diffusivity. The subscripts 0 and 1 on concentrations refer to the surface location and the first residence site beneath the surface

3.3 Movement Through Structures

Movement of solute gas atoms in a structure is governed by the conservation of atoms of species “s” described by

$$\frac{\partial C_s}{\partial t} = -\nabla \cdot J_s + S_s - \sum_k \left(\frac{\partial C_s^{t_k}}{\partial t} + \nu_s C_s^{t_k} \right) - \nu_s C_s + \sum_m \nu_m^s \left(C_m + \sum_k C_m^{t_k} \right) \quad (31)$$

where

C_s = mobile concentration of species “s” atoms in the structure

J_s = diffusive flux of species “s” atoms

S_s = local explicit source rate of species “s” atoms

$C_s^{t_k}$ = concentration of atoms of species “s” in the “kth” trap

ν_s = radioactive decay frequency of species “s” atoms

ν_m^s = radioactive decay frequency of species “m” atoms that decay to species “s”

The most general statement about flows in a thermodynamic system is

$$J_q = \sum_r L_{qr} F_r \quad (32)$$

where

J_q = generalized flow involving some extensive thermodynamic variable

L_{qr} = generalized kinetic coefficient

F_r = generalized force, usually the gradient of the intensive thermodynamic variable conjugate to J_q .

When the extensive property involved in the flow is moles of dissolved gas, the conjugate force is the gradient in the concentration of that species of gas atoms, the kinetic coefficient is the negative of the diffusivity, and the extensive flow is the diffusive flux. However, because gradients in other intensive properties are also likely to exist and there may be coupling between these forces and non-conjugate flows, there may be more terms than one in the sum. All versions of TMAP have considered only temperature gradients in addition to the concentration gradients because the kinetic coefficients for most other forces (e.g., gradients in electric potential and magnetization) will generally be small. Thus the flux of

dissolved gas atoms is given in TMAP7, as in previous versions, by Equation (31) with the supplementary relation

$$\nabla \cdot \mathbf{J}_s = \frac{d}{dx} \left[-D \left(\frac{dC_s}{dx} + C_s \frac{Q^*}{kT^2} \frac{dT}{dx} \right) \right] \quad (33)$$

where D is the diffusivity of species s in the matrix, Q^* is the heat of transport or Soret coefficient, and T is the local temperature.

The inventory of solute gas atoms (or molecules) can be affected by trapping sites where the energy required for an atom to move to another nearby site is greater than that required for ordinary diffusion. Trapping sites can result from impurities and structural irregularities arising from cold work, precipitation of alternate phases including gas bubbles, or neutron displacement damage. Considering a solute particle (atom or molecule) concentration of species “s”, C_s , the governing equations are written with respect to the “kth” trap as

$$\frac{\partial C_s^{t_k}}{\partial t} = \frac{\alpha_{t_s} C_s^e}{N} C_s - (\alpha_{r_k} + \nu_s) C_s^{t_k} \quad (34)$$

$$C_{t_k}^e = C_{t_k}^o - \sum_s C_s^{t_k} \quad (35)$$

$$\alpha_{t_s} = \frac{D_s}{\lambda^2} \quad (36)$$

$$\alpha_{r_k} = \nu_o \exp \left(-\frac{E_{t_k}}{kT} \right) \quad (37)$$

Here, in addition to previously defined quantities,

$C_s^{t_k}$ = concentration of atoms of species “s” in the “kth” trap

$C_{t_k}^e$ = concentration of empty traps of type “k”

$C_{t_k}^o$ = total concentration of traps of type “k” both filled and unfilled

E_{t_k} = trapping energy of the “kth” trap

α_{t_s} = trapping rate coefficient of species “s” atoms

α_{r_k} = release rate coefficient for trapped atoms of any species from trap “k”

Note in the above that different mobile diffusion species will have different trap frequencies (capture rates) into a given trap. For hydrogen isotopes, for example, heavier isotopes diffuse more slowly. Therefore, they fall into traps at a lower frequency than their lower mass siblings. However, the depth of the trap would have no particular effect on the trapping rate. Conversely, because the Boltzmann factor for escape from the trap is only energy dependent, it would be independent of the species in the trap (assuming equipartition of energy), but it will depend on the trap strength.

An implicit assumption in this model is that there is a temporally fixed number of trap sites of each kind in a given volume, though the concentration of such sites may be spatially distributed. Treatment of trapping in TMAP7 is stochastic. Trap and trapped atom concentrations are considered intensive properties.

3.3 Structural Thermal Response

The processes that govern the permeation of solute gas atoms in structures are highly temperature dependent. Therefore, TMAP7 was developed with the capability of solving the time-dependent heat conduction equation to determine the thermal response of a structure. This equation is

$$\rho C_p \frac{\partial T}{\partial t} = \nabla \cdot (\kappa \nabla T) + S_h \quad (38)$$

where

- ρ = material mass density
- C_p = material specific heat
- T = temperature
- κ = material thermal conductivity
- S_h = local volumetric heating rate.

This equation is solved in conjunction with the boundary conditions of either surface convection, an imposed surface heat flux, a specified temperature, or an adiabatic surface.

The thermal response calculations are extended for a linked-segment (composite material) structure by the use of a gap or surface-contact conductance equation at the interface. The interface equation for heat flow across the interface between the two structures is

$$q = h_{gap} (T_{s2} - T_{s1}) \quad (39)$$

where

q = interface heat flux

h_{gap} = user defined gap thermal conductance

T_{s_2} = surface temperature of the segment more positive in the distance coordinate

T_{s_1} = surface temperature of the segment more negative in the distance coordinate.

Here, segment “2” is positioned further in the positive “ x ” direction than is segment “1”.

3.3.1 Surface Boundary Conditions

TMAP7 provides four types of thermal boundary conditions: **adiabatic**, **convective**, **set-surface-temperature**, and **surface-heat flux**. Each condition incorporates a different equation into the thermal tridiagonal matrix equation set.

For the adiabatic (*adiab*) boundary condition, the temperature at the surface of the structural segment is made equal to the temperature at the first node inside the surface. That forces the thermal gradient to zero, thereby preventing heat flux.

The convective (*conv*) boundary condition equates the conductive flux at a surface with the convective heat transfer coefficient, h , and a temperature difference, $T_f - T_e$, where T_f is the surface temperature and T_e is the adjacent enclosure temperature. The finite difference equation for the boundary is then

$$\left[h + \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1} \right) \right] T_f - \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1} \right) T_1 = h T_e \quad (40)$$

where the values of κ are values of thermal conductivity at the various nodes, and the subscript 1 refers to the first internal node.

The user has the option of a set-surface-temperature (*stemp*) boundary condition in which the temperature of the surface node of the segment is specified. The temperature specification may be in the form of a constant, an equation, or a look-up table.

For the specified-heat-flux or *sflux* boundary condition, an expression is provided by the user for the heat flux, q_h , at the surface of the segment. This expression may be a constant, an equation, or a time-dependent look-up table. The equation to be solved for the boundary is

$$\left(\frac{\kappa_1 + \kappa_f}{\Delta x_1} \right) T_f - \left(\frac{\kappa_1 + \kappa_f}{\Delta x_1} \right) T_1 = q_e \quad (41)$$

3.3.2 Composite Structure Treatment

When two elements of a composite structure are in thermal contact, there is frequently an insulating layer, such as an oxide layer, that inhibits heat transfer, which in metals is often dominated by electron conductivity.⁷ Also, there may be an actual gap between the segments for which it is possible to specify a thermal conductivity. In either of these cases, the temperature at one side of the thermal interface will not necessarily be the same as the temperature at the other side of that interface. However, of necessity the heat flux at the composite interface must be conserved. The expression for that continuity is

$$-\kappa \frac{\partial T}{\partial x} \Big|_{f^A} = -\kappa \frac{\partial T}{\partial x} \Big|_{f^B} = h_{gap} (T_f^A - T_f^B) \quad (42)$$

The relationship is illustrated in Figure 2. The subscript “f” refers to values at the surface of the structural segment. The superscripts “A” and “B” refer to segments of the different materials on the two sides of the gap. The resulting equation in tridiagonal form for surface “A” is

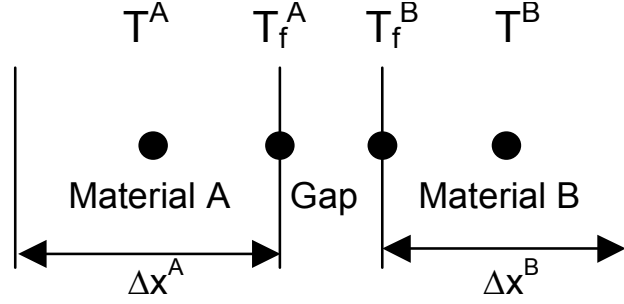


Figure 2. Relationship of quantities for heat flux at the interface of a composite material.

$$-\left(\frac{\kappa_f^A + \kappa^A}{\Delta x^A}\right)T^A + \left[\left(\frac{\kappa_f^A + \kappa^A}{\Delta x^A}\right) + h_{gap}\right]T_f^A - h_{gap}T_f^B = 0 \quad (43)$$

and there is a corresponding relation for the “B” surface.

3.4 Enclosure Response

TMAP7 allows two types of user-defined enclosures, “functional” and “boundary” types. If we consider the “ith” enclosure and apply the finite differencing technique, enclosure dynamics for *functional* type enclosures are governed by

$$V_i \frac{C_{m_i}^n - C_{m_i}^o}{\Delta t} = \sum_j C_{m_j}^n Q_{ji} - C_{m_i}^n \sum_j Q_{ij} - \sum_k A_k J_{m_k} + V_i S_{chem_m} \quad (44)$$

This says that the rate of increase of the concentration of gas molecular species “m” in enclosure “i” is the sum of flows of that species into that enclosure from other connected enclosures, less the flow of that species from the “ith” enclosure to the other enclosures, less that flowing into the adjacent “k” surfaces, plus that generated in the enclosure volume by chemical reactions. That source term includes all reactions that involve species m.

Unlike most other state variable equations, this equation is solved using the Newton Raphson iteration technique. That is, for any time step “τ”

$$C_{m_i}^{\tau+1} = C_{m_i}^{\tau} - \omega \frac{F}{\frac{\partial F}{\partial C_{m_i}}} \quad (45)$$

where ω is a relaxation parameter to optimize convergence. For this application, the function “F” is defined as

$$F = V_i \left(\frac{C_{m_i}^v - C_{m_i}^o}{\Delta t} \right) - \sum_j C_{m_j}^\tau Q_{ji} - C_{m_i}^\tau \sum_j Q_{ij} - \sum_k A_k J_{m_k} + V_i S_{chem_m} \quad (46)$$

and

$$\frac{\partial F}{\partial C_m} = \frac{V_i}{\Delta t} + \sum_j Q_{ji} - \sum_k A_k \frac{\partial J_{m_k}}{\partial C_m} - V_i \frac{\partial S_{chem_m}}{\partial C_{m_i}} \quad (47)$$

For the chemical rate derivative, we have discretized the chemical derivative term as

$$\frac{\partial S_{chem_m}}{\partial C_{m_i}} = \sum_j \left(-a_{m_j} + b_{m_j} \right) \left(\frac{\tilde{R}_j - R_j^\tau}{\tilde{C}_{m_j} - C_{m_i}^\tau} \right) \quad (48)$$

where $\tilde{C}_{m_i} = 1.05 C_{m_i}$ is an arbitrary concentration offset defined solely for the purpose of generating approximations to derivatives. \tilde{R}_{m_i} is the forward reaction rate evaluated when $C_{m_i}^\tau = \tilde{C}_{m_i}^\tau$. The subscript “ m ” denotes the different molecular species present. After convergence, terms with the “ $\tau + I$ ” superscript become new time values with superscript “ τ ”.

The relations for the molecular fluxes into surfaces and their derivatives depend on the type of boundary condition imposed for the diffusion segments.

For **ratedep** boundary conditions, we have from Equation (2) for molecular flux into the “ kth ” surface

$$J_{m_k} = K_{d_m} P_m - \sum_{i=1}^n K_{r_{ij}} C_{i_k} C_{j_k} \quad (49)$$

Then with the relation

$$P_{m_i} = C_{m_i} k T_{e_i} \quad (50)$$

where

k = Boltzmann’s constant

T_{e_i} = temperature of enclosure “ i ”

we see that

$$\frac{\partial J_{m_k}}{\partial C_{m_i}} = K_{d_{m_k}} k T_{e_i} \quad (51)$$

If the **lawdep** condition applies, where the solute concentration on the surface is related to the partial pressure of the gas above the surface through a solubility law such as Sieverts' law, then Equations (23) apply in defining the constituent flows to the surface. To calculate the constituent flows to the surface,

$$J_{s_f} = K_\lambda \left(C_m^\tau k_B T_e \right)^\varepsilon (g_{t2} - g_2) + C_{s_1}^\tau (g_{t2} + g_2) \quad (52)$$

where

K_λ = the appropriate solubility coefficient (see Equations(18) and (27))

$$g_2 = \frac{D_{s_f} + D_{s_1}}{\Delta x_1} \quad (53)$$

$$g_{t2} = \frac{Q_{s_f}^* + Q_{s_1}^*}{R(T_f - T_1)^2} (T_1 - T_f) g_2, \quad (54)$$

For the Newton-Raphson iteration derivative of Equation(47), we find

$$\frac{\partial J_{s_k}}{\partial C_m} = \varepsilon K_\lambda (k_B T_e)^\varepsilon C_m^{\varepsilon-1} (g_{t2} - g_2). \quad (55)$$

Once the flows to the surface of constituents (e.g., H, D) have been found using the assumption that they interact only with homonuclear species (e.g., H₂, D₂), and species addition by convective flow or chemical reactions has been determined, species equilibration to find the heteronuclear species concentrations is computed. Recognizing that the total concentrations of types *A* and *B* particles in the enclosure is

$$\begin{aligned} C_A &= C_{AB} + \frac{C_{A_m}}{\nu_A} \\ C_B &= C_{AB} + \frac{C_{B_n}}{\nu_B} \end{aligned} \quad (56)$$

we see that

$$\begin{aligned} C_{A_m} &= (C_A - C_{AB}) \nu_A \\ C_{B_n} &= (C_B - C_{AB}) \nu_B \end{aligned} \quad (57)$$

For equilibrium to hold, the equilibrium (primed) value of C_{AB} must be, from Equation (26)

$$C'_{AB} = \eta [(C_A - C_{AB})\nu_A]^{\nu_A} [(C_B - C_{AB})\nu_B]^{\nu_B} \quad (58)$$

That is solved for the equilibrium heteronuclear species concentration, C'_{AB} , using the bisection method of successive approximations and the values of C_A and C_B found using Equations (56). Then the homonuclear concentrations are found from C'_{AB} using Equations (57) and the values of C_A and C_B , which have not changed.

If we specify the surface concentration directly in the **sconc** boundary condition, we assume no coupling between the molecular gas species in the enclosure and any flux of atoms to the surface of the material. Hence, the “ J_{m_k} ” term drops out of the Newton-Raphson function “ F ”. The same thing holds if the **nonflow** boundary condition is applied to the surface(s) bounding the enclosure.

4.0 INPUT FILE

In the material that follows, instructions are provided for the various entries in the TMAPINP file used with TMAP7. Words in **bold** typeface are keywords that need to be entered in lower case as shown. Generally, spaces are optional. Items shown in angled brackets (<>) are input data. Those shown with (α) are alphanumeric constants, those with (*i*) are of type integer while those with (*r*) are real or floating-point numbers. Even though the code operates fully in double precision arithmetic for real numbers, the input preprocessor does not recognize "d" notation. Exponents in scientific notation should be entered with a lower-case "e".

Most statements are required unless otherwise indicated. Those that are optional are included in brackets ([]). Note that nearly all statements are terminated with "**end**". This is a key to the preprocessor that the statement is concluded. Multiple line entries are thus permitted without any continuation indicator. Where multiple parameters are separated by commas in a statement requiring more than one line, no comma should be placed at the end of a continued line nor at the beginning of the next one. That function is provided by the carriage return marking the end of the line. Input file statements may fill the first 80 columns of any line. Comment statements must begin with the dollar sign (\$). These may not appear in the middle of a multiple-line statement, though they may appear on the same line following the "**end**" of a statement.

We now proceed to give the statement specifications, generally in the order they would be entered in the TMAPINP file. Examples are provided at the end of each input block description.

4.1 Title Input

The only statement in the *title input* block is the title input statement. The title may be up to 5 lines of 80 characters, beginning in column 1.

title input

Example:

```
title input
  Sample Problem #5 - Co-Permeation of D and H through Pd by K. Kizu,
  A. Pisarev and T. Tanabe, Journal of Nuclear Materials, 289 (2001) 291-302.
end of title input
```

4.2 Main Input

The *main input* block defines the various species types, the numbers of segments in the various structures, and the number of enclosures in the problem. Diffusion species are entered using the following statement.

dspcnme = <dspc(α)₁>,<dspc(α)₂>,...,<dspc(α)_n>,end

This statement provides names to each of the diffusing species. The number of names given is the number of diffusing species in the problem. This list does not include those species to be

separately identified as surface-only species. The maximum number of diffusing species is 10. Diffusion species names are limited to 8 characters.

[dkrate = <dspc(α)_{s1}>, <dcay(r)₁>, [<dspc(α)_{p1}>]

[, <dspc(α)_{sn}>, <dcay(r)_n>[, <dspc(α)_{pn}>],] . . . end]

This optional statement is the input for radioactive decay. It must follow the name declaration statement just discussed. The first entry following the “=” sign is the *dspcnme* of the species that decays. It must be one of the names entered in the previous statement as a diffusion species. The second entry is the decay frequency for the species that is decaying or the reciprocal of its mean lifetime in seconds. The third entry is optional and identifies the product of the decay. If it is entered, it must be one of the diffusion species *dspcnmes*. In that case, an increase of that species will be observed as the decaying species decays. If the product species name is omitted, the decaying species simply vanishes as it decays. Multiple groups of two or three statements may be entered, one for each species that decays. However, if no product *dspcnme* is given in the leading group, there must be a double comma before the decaying species *dspcnme* in the following one.

espcnme = <espc(α)₁>, <espc(α)₂>, ..., <espc(α)_n>, end

This statement is where enclosure species names are defined. The number of enclosure species allowed is 30. Enclosure species names are also limited to 8 characters. They may replicate diffusion species names, such as when Henry's law applies, but they are considered as separate species within the code, and repeating the names can lead to confusion.

[sspcnme = <sspc(α)₁>, <sspc(α)₂>, ..., <sspc(α)_n>, end]

This is an optional statement that identifies surface-only species. These species allow expansion of calculational possibilities that may include such things as the formation of molecules that remain attached to the surface and interact with diffusing species at the surface without actually diffusing themselves. Examples may be water or methane and the complexes involved in their formation such as hydroxyl radicals. Rules for surface species interaction with enclosure species depend on the enclosure type and the diffusion boundary conditions at the surface. Surface-only species are only meaningful under *surfdep*, and *ratedep* diffusion boundary conditions. For *lawdep*, *sconc* and *nonflow* boundary conditions, surface species are meaningless since diffusion species concentrations do not depend on them. Further, for *lawdep* boundary conditions interacting with functional enclosures, concentrations of heteronuclear species in the enclosures must be computed using an equilibrium law. These various boundary condition types are summarized in Table 1.

Table 1. Summary of applicability and treatment of surface species for various combinations of enclosure type and diffusion boundary conditions.

Diffusion Boundary Condition	Functional Enclosure	Boundary Enclosure
<i>Surfdep</i>	Captured by species arrival from gas; returned to gas by escape from the surface; no species changes enroute, means there must be an enclosure species corresponding to each surface species that can escape or be captured; rates dynamically computed.	Captured by species arrival from gas; returned to gas by escape from the surface; no species changes enroute, means there must be an enclosure species corresponding to each surface species that can escape or be captured; rates of exchange are irrelevant because enclosure species concentrations are pre-specified..
<i>Ratedep</i>	Undergo dissociation and recombination reactions with automatic release to enclosure species (similar to diffusion species), but also surface formation, dissociation ^a , and combination reactions; rates to enclosure are dynamically computed	Undergo dissociation and recombination reactions with automatic release to enclosure species (similar to diffusion species), but also surface formation, dissociation ^a , and combination reactions; rates irrelevant because enclosure species concentrations are pre-specified.
<i>Lawdep</i>	Meaningless because surface concentrations of diffusion species are set by assigned solubility relations.	Meaningless because surface concentrations of diffusion species are set by assigned solubility relations.
<i>Sconc</i>	Meaningless because diffusion species concentrations are fixed, not subject to adjustment by surface interactions.	Meaningless because diffusion species concentrations are fixed, not subject to adjustment by surface interactions.
<i>Nonflow</i>	Meaningless because surface is not available for reactions.	Meaningless because surface is not available for reactions.

^aConventional dissociation/recombination reactions associated with the *ratedep* boundary condition are those relating surface concentrations of the species in question to the gas molecules above the surface. Surface formation and dissociation reactions addressed here involve other species at the surface. Formation means forming the surface species from surface constituents while dissociation is the reverse. Combination is the incorporation of the subject surface species with another one into a third.

For functional enclosures and *surfdep* diffusion boundary conditions, both surface-only and diffusion species may desorb if they can overcome the surface binding potential, ($E_x - E_c$ in Figure 1), but for most physical situations they come off only by formation of a larger molecule having a lower surface binding energy. Surface-only species must have names different from

those in the diffusion species name list. A maximum of 30 surface species names may be defined.

To illustrate the need for so many, if all three hydrogen isotopes were present as diffusion species, and we considered only oxygen additionally, we would need 3 diffusion species (H, D, T), 3 surface species (OH, OD, OT), and 13 enclosure species (6 forms of hydrogen gas, 6 forms of water and O₂). If ammonia formation were added, we would need 12 additional enclosure species and 10 additional surface species.

segnds = <nodes(i)₁>, <nodes(i)₂>, ..., <nodes(i)_n>, end

The number of nodes is given here for each of the segments in the problem. The number of entries provided is the number of *thermseg/diffseg* segments. The maximum number of *thermsegs/diffsegs* is 50 with a maximum number of nodes, including surface nodes, of 1,000.

nbrencl = <nbr(i)>, end

This statement defines the number of enclosures in the problem. The maximum number of enclosures is 40.

[linksegs = <seg(i)₁>, ..., <seg(l)_{n1}>[, also, <seg(l)_{n1+1}>, ... <seg(l)_{n2}>], end]

This statement defines segment linking. There is no arbitrary limit to the number of segments that can be linked together along any particular path, so long as the number of total segments does not exceed 50. Note that in TMAP7 the segment linking need not be in numerical sequence.

Example:

```
main input
  dspcnme=h,t,he,end
  espcnme=h2,t2,ht,end
  segnds=12,18,end
  linksegs=1,2,end
  nbrencl=5,end
  dkrate=t,1.782411e-9,he,end
end of main input
```

4.2 Enclosure Input

In this section the user provides TMAP7 specific data required to define the *enclosure* control volumes. There are two types of enclosures. One is called *functional*, and the other is a *boundary* enclosure. *Functional* enclosures allow dynamic pressure and chemical reaction calculations to be included during the running of the problem. There may be diffusive flows of enclosure species to and from the surfaces. There may also be convective flows from one functional enclosure to another or to and from *boundary* enclosures. On the other hand, *boundary*-type enclosures are associated with pre-determined conditions that do not depend on thermal effects or mass diffusion. Neither convective fluid flows nor diffusive enclosure species

flows to and from boundary type enclosures are computed because properties in boundary enclosures are pre-specified. They may be used as sources and sinks for functional enclosures, however, where those flows are computed.

Each enclosure is identified with a separate sub block of data. The sequence in which these definitions are made determines the enclosure relative sequence number. However, any enclosure reference number may be defined by the user provided that each is unique and that all the numbers between 1 and the number of enclosures defined are used. The sequence of entering enclosure types and the numbers assigned to the enclosures are otherwise arbitrary. This is a change from previous code versions.

Each sub-block begins with a start declaration statement, but there is no terminal declaration for the sub-block. There are order-independent parameter statements after the start statement for each sub-block of enclosure inputs. The number of these statements depends on whether the enclosure is *functional* or *boundary* in nature. We first consider *functional* enclosures.

4.2.1 Functional Enclosures

There are five parameter statements following the "start" declaration in specifying *functional* enclosures. Of these, two are optional: the specification of fluid flows between enclosures and that for chemical reactions.

start func,<enclnbr(i)>[,end]

This statement begins the functional enclosure definition sub-block. The <enclnbr(i)> value entered is the assigned integer enclosure number.

$$\text{etemp} = \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{end}$$

This statement identifies the temperature of the enclosure. The value of that temperature may be a constant, indicated by entering just the number or the keyword **const** and the number, or it may be defined in terms of an equation or a lookup table as a function of time. This temperature is used with the enclosure species concentrations to calculate partial pressures of the various enclosure species. In previous code versions, enclosure temperature was fixed. Temperatures are entered in kelvins.

esppres = <espc(α)₁>,<value(r)₁>,...,<espc(α)_n>,<value(r)_n>,end

This statement enters initial partial pressures for the various enclosure species. There must be an entry for each of the enclosure species identified in the *main input* block. These are then changed by diffusive and convective fluid flows during the course of the problem. Pressures are entered in pascals.

When a functional enclosure faces a diffusion segment under *lawdep* boundary conditions, only one enclosure species can be linked to the concentration of any diffusion species. Normally, a hydrogenic or other atomic diffusion species, whose gaseous state is diatomic, would be related to the partial pressure of its diatomic homonuclear molecule. The *lawdep* boundary condition implies chemical equilibrium at the surface, so if there is more than one diffusion species, the specified enclosure species may dissociate to atoms that combine to form heteronuclear molecules (e.g., HD). In these instances, heteronuclear molecular pressures will need to be calculated automatically by including the following statement.

$$\begin{aligned}
 &[\text{espcomb} = \langle \text{espc}(\alpha)_1 \rangle, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\} \\
 &\quad \langle \text{espc}(\alpha)_2 \rangle, \langle \text{stoich}(r)_2 \rangle \langle \text{espc}(\alpha)_3 \rangle, \langle \text{stoich}(r)_3 \rangle \\
 &\quad \left[\text{also}, \langle \text{espc}(\alpha)_1 \rangle, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\} \right. \\
 &\quad \left. \langle \text{espc}(\alpha)_2 \rangle, \langle \text{stoich}(r)_2 \rangle \langle \text{espc}(\alpha)_3 \rangle, \langle \text{stoich}(r)_3 \rangle, \dots \right], \text{end}]
 \end{aligned}$$

This optional statement allows the user to specify which of the enclosure species combine together at the surface to make a third when at least one diffusion segment facing this *functional* enclosure has the *lawdep* diffusion boundary condition. Examples are H₂ and D₂, which combine at the surface to make HD, or H₂ and O₂, which combine to form H₂O. Under these conditions, where homonuclear molecular gas pressures are specified, the code will automatically calculate corresponding heteronuclear molecular pressures according to

$$P_C = \eta P_A^{\nu_A} P_B^{\nu_B} \quad (59)$$

In the *espcomb* statement, $\langle \text{espc}(\alpha)_1 \rangle$ is the name of the heteronuclear enclosure species associated with P_C and similarly $\langle \text{espc}(\alpha)_2 \rangle, \langle \text{espc}(\alpha)_3 \rangle$ are names of the specified homonuclear species P_A and P_B . The value following $\langle \text{espc}(\alpha)_1 \rangle$ is the equilibrium constant η which may be assigned values as a constant, or as an equation or lookup table. The table is assumed to have temperature as the independent variable, while the equation form can specify either time or temperature or both as independent variables. The $\langle \text{stoich}(r)_i \rangle$ are the stoichiometric coefficients ν_i . Only those species identified in this statement will be automatically computed. Note that only one specification for each heteronuclear molecule so formed should be given, and only binary reactions are facilitated in the code.

This specification is for the current functional enclosure only. One must be provided for other enclosures where needed. If, in addition to at least one diffusion segment having the *lawdep*

diffusion boundary condition, there are other diffusion segments facing this enclosure that have other diffusion boundary conditions, the heteronuclear molecular species will still be calculated and used.

[reaction = nequ,<neq(i)>

ratequ,<eqnbr(i)>

nreact,<nbrreact(i)>,<espc(α)₁>,<stoich(r)₁>,...,

<espc(α)_n>,<stoich(r)_n>

nprod,<nbrprod(i)>,<espc(α)₁>,<stoich(r)₁>,...,

<espc(α)_n>,<stoich(r)_n>

:

[ratequ,<eqnbr(i)>

nreact,<nbrreact(i)>,<espc(α)₁>,<stoich(r)₁>,...,

<espc(α)_n>,<stoich(r)_n>

nprod,<nbrprod(i)>,<espc(α)₁>,<stoich(r)₁>,...,

<espc(α)_n>,<stoich(r)_n>]

end]

This statement defines chemical reactions taking place in the enclosure volume. These are computed as if the enclosure were a static volume with no flows. The chemical reaction specified in the statement is represented by



An example is the reaction $\text{H}_2 + \frac{1}{2} \text{O}_2 \rightarrow \text{H}_2\text{O}$ where a is 1, A is H_2 , and b is $\frac{1}{2}$, B is O_2 , c is 1, C is H_2O , and d and D vanish. The rate equation called for is for the forward reaction or the frequency (s^{-1}) with which a molecules of A and b molecules of B are consumed in the reaction and at which c molecules of C and d molecules of D are produced. The forward reaction rate is

$$R_f = K_f [A]^a [B]^b \quad (61)$$

The implicit assumption in this definition is that the reaction goes to completion. If the reaction is an equilibrium reaction, then the reverse reaction should also be defined; i.e.,

$$R_b = K_b [C]^c [D]^d \quad (62)$$

This can be accomplished in a single rate equation with an appropriate reaction rate definition. Note that in equilibrium

$$\begin{aligned} R_f &= R_b \\ K_f [A]^a [B]^b &= K_b [C]^c [D]^d \\ \frac{[C]^c [D]^d}{[A]^a [B]^b} &= \frac{K_f}{K_b} = K_{eq} \end{aligned} \quad (63)$$

which is the familiar law of mass action for chemical reactions. Thus, an appropriate forward rate equation for the equilibrium reaction referred to here may be

$$R = K_f \left([A]^a [B]^b - \frac{1}{K_{eq}} [C]^c [D]^d \right) \quad (64)$$

When constructing the equations for R_f and/or R_b , which are the rate equations referred to in the *reaction* statement, it is usually necessary to reference the concentrations of the appropriate reactants. These are available in the subroutine that evaluates the equations under the name **conce**(*i*), where *i* is evaluated from

$$i = (\text{relative enclosure number} - 1) \times (\text{number of enclosure species}) + j \quad (65)$$

where *j* is the sequence number associated with the enclosure species to be referenced. That number is its relative position in the list supplied in the *espcnme* statement in the *main input* data block, the first species listed being number 1, the second number 2, and so forth. For example, suppose enclosure species *A*, *B*, and *C* had been identified in the main input block and there were three enclosures. To reference the concentration of *B* in the third enclosure defined, *j* would be 2, and one would refer to **conce**(8) in the rate equation.

In the *reaction* statement, **nequ** is a keyword identifying the following integer, **<neq(i)>**, as the number of chemical reactions defined in the current functional enclosure. A sub-block must be defined for each of the reactions beginning with the keyword, **ratequ** and the integer number **<eqnbr(i)>** corresponding to the sequence number of the rate equation in the *equation input* list to follow later. Then following the keyword **nreact** are listed the enclosure species names and stoichiometric coefficients for each of the species on the left side of the reaction equation. A corresponding list of species names and stoichiometric coefficients follows the keyword **nprod** for the reaction products, those on the right side of the reaction equation. The **end** keyword terminates the statement after each of the **ratequ** sub-blocks has been entered.

evol = <enclv(r)>,end

This statement defines the volume of the functional enclosure. Units are cubic meters.

[outflow = nbrflwp,<npath(i)>

qflow, $\left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{rencl}, < \text{value}(i) >$

:

[qflow, $\left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{rencl}, < \text{value}(i) >, \text{end}]$

This optional statement defines the parameters of the flows from the present enclosure to other enclosures. The integer **<npath(i)>** following the keyword **nbrflwp** is the number of flow paths that are defined *leaving* the functional enclosure. This does not include flow paths coming into the enclosure from elsewhere. The definition of each flow path begins with the keyword **qflow**. Each flow rate is defined with a value, which may be preceded with the keyword **const**, or with an equation (keyword **equ** followed by the equation number) or with a table (keyword **tabl** followed by the table number). Flow rates are entered in m³/s. Following the keyword **rencl**, the enclosure number of the *receiving* enclosure is identified. This is the user assigned enclosure number, not necessarily the same as the sequence in which it is defined. The receiving enclosure is the one into which the flow from the present enclosure is going. It may be either a functional or a boundary enclosure.

4.2.2 Boundary Enclosures

Boundary enclosures represent pre-specified environments that are not altered by convective flows or diffusion effects. Two parameter statements are required for *boundary* enclosures following the "start" declaration for the sub-block, and one more is optional.

start bdry,<enclnbr(i)>[,end]

This statement identifies the beginning of information for a boundary enclosure to which the number **<enclnbr(i)>** is to be assigned.

$$\text{etemp} = \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{end}$$

Temperatures are allowed to change in boundary enclosures as a function of time. Real values input alone or with the keyword **const** are the constant temperature over the duration of the problem. Integers entered after keywords **equ** or **tabl** are equation or table numbers, respectively, for evaluating the enclosure temperature, again, as a function of time. Temperatures must be entered in kelvins.

$$\text{esppres} = <\text{espc}(\alpha)_1>, \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \dots,$$

$$<\text{espc}(\alpha)_n>, \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{end}$$

This statement provides a specification, entered in pascals, for the enclosure species partial pressures in *boundary* enclosures. If the diffusion boundary conditions for diffusion segments connected with this enclosure are *surfdep* or *ratedep*, then an entry should be made for each enclosure species (**<espc(α)>**) present. On the other hand, if the diffusion boundary conditions are *lawdep*, the user should only provide entries for one enclosure species per diffusion species, normally the homonuclear molecular species (e.g., H₂, D₂). For boundary enclosures, these pressures may be constant or they may vary in time according to an equation or a table. They do not depend on any flows or other parameters.

[outflow = nbrflwp, <npath(i)>

$$\text{qflow}, \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{rencl}, <\text{valuer}(i)>$$

$$\vdots$$

$$[\text{qflow}, \left\{ \begin{array}{l} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}, \text{rencl}, < \text{value}(i) >, \text{end}]$$

This statement is the same for both functional and boundary enclosures.

Example:

```
enclosure input
$
start bdry,1,end
$ This is the background pressure source for both active chambers
  etemp=tabl,1,end
  espres=h2,1.0e-5,hd,2.0e-5,d2,1.0e-5,end
  outflow=nbrflwp,2,qflow,const,0.1,rencl,2
  qflow,const,0.1,rencl,3,end
$
start func,2,end
$ This is the upstream chamber connecting to the membrane
  etemp=tabl,1,end
  espres=h2,1.e-5,d2,1.e-5,hd,2.e-5,end
  outflow=nbrflwp,1,qflow,const,0.1,rencl,4,end
  evol=0.05,end          $ Estimated volume
  espcomb=hd,const,2.0,h2,0.5,d2,0.5,end
$
start func,3,end
$ This is the downstream chamber connected to the membrane
  etemp=tabl,1,end
  espres=h2,1.e-5,d2,1.e-5,hd,2.e-5,end
  outflow=nbrflwp,1,qflow,const,0.1,rencl,4,end
  evol=0.05,end          $ Estimated volume
  espcomb=hd,const,2.0,h2,0.5,d2,0.5,end
$
start bdry,4,end
$ This is the gas sink representing the vacuum pumping system
  etemp=tabl,1,end
  espres=h2,1.e-10,hd,1.e-10,d2,1.e-10,end
  outflow=nbrflwp,2,qflow,const,0.1,rencl,2
  qflow,const,0.1,rencl,3,end
$
start bdry,5,end
$ This is the gas source with pre-programmed species pressures
  etemp=tabl,1,end
  espres=h2,const,1.4e-1,hd,const,1.0e-10,d2,tabl,2,end
  outflow=nbrflwp,1,qflow,const,0.1,rencl,2,end
end of enclosure input
```

4.3 Thermal Input

There must be a *thermseg* sub-block entry for each individual diffusion/heat-transfer element, whether or not it is linked. It is also the place where the geometric structure of the

thermseg/diffseg is defined.

start thermseg [,end]

This statement begins each sub-block defining heat transfer conditions. The “**end**” closure is optional on this statement. The order of statements within the sub-block is optional. The order in which the various *thermsecs/diffsecs* is defined gives them a relative sequence number. Their order in this block must agree with the order in which the numbers of nodes for these segments were defined in the “**segnds**” statement in the *main input* block. All references to them make use of the code-assigned sequence identifier. Partial comment lines, beginning with “\$” after the “**end**” statement, or full comment lines may be used for clarity.

delx = 0.0,<dx(r)₁>,<dx(r)₂>, . . . ,<dx(r)_n>,0.0,end

This required statement defines the nodal separations (Δx_i) for this *thermseg/diffseg*. The real-number entries, $\langle dx(r)_i \rangle$, with units of meters, are identified with the nodes in left-to-right order, that is, in increasing distance, x. Both left- and right-surface nodes are required to have zero thickness, even if the segments are linked. There must be an entry for each node, and the number of entries, including one for each surface node, must correspond to the numbers defined in the **segnds** statement in the *main input* data block.

Arbitrary subsets of the input data having the same numerical value may be combined in a shorthand input format:

. . . <numnds(i)>*<dx(r)>. . .

where $\langle numnds(i) \rangle$ is an integer indicating how many nodes are to be specified in this cluster, and $\langle dx(r) \rangle$ defines the uniform width of each of those segments. An example of such a specification may be "6*3.0e-4".

The statement defining initial temperature distribution within the *thermseg/diffseg* is:

$$\text{temp} = \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle \\ \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle \\ \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle \end{array} \right\}, \text{end}$$

This new specification for initial temperature distribution in the structure has the added flexibility of applying functional distributions for temperature. A series of simple numerical values may be entered (one for each node or combined using the $\langle numnds(i) \rangle * \langle \text{value}(r) \rangle$ format). Alternatively the following may be entered:

temp = const,<value(r)₁>

This sets the temperature value at every node equal to $\langle \text{value}(r)_1 \rangle$.

$$\text{temp} = \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle$$

This form provides the temperature at each node as a linear interpolation between $\langle \text{value}(r)_1 \rangle$ at the left face and $\langle \text{value}(r)_2 \rangle$ at the right face of the *diffseg*.

$$\text{temp} = \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle$$

This form calculates the initial temperature at each node as a decaying exponential beginning at the left face as $\langle \text{value}(r)_1 \rangle$ and decaying to $\langle \text{value}(r)_2 \rangle$ with a $1/e$ width of $\langle \text{value}(r)_3 \rangle$ (m). If $\langle \text{value}(r)_2 \rangle$ is greater than $\langle \text{value}(r)_1 \rangle$, the function is a saturating exponential.

$$\text{temp} = \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle$$

This form calculates the temperature as the sum of a normal distribution with a maximum value of $\langle \text{value}(r)_1 \rangle$ centered at a distance x_0 equal to $\langle \text{value}(r)_2 \rangle$ (m) from the left face, and having a characteristic width of $\langle \text{value}(r)_3 \rangle$ (m); combined with a constant background value of $\langle \text{value}(r)_4 \rangle$.

These functional forms may have greater or less usefulness for temperature, but they are also used for specifying initial species and trap concentration distributions, as discussed below.

$$[\text{tcon} = \left\{ \begin{array}{l} \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{end}]$$

This statement is optional but it must be included if material properties within the segment are to vary with time and/or temperature. It defines the thermal conductivity in W/m-K for the entire segment. There is no provision for this material property to vary spatially within an individual segment, except as temperature changes spatially. The value is specified either as a constant with value $\langle \text{value}(r) \rangle$; as an equation, which may be a function of “time” and/or “temp”, which appears as equation $\langle \text{value}(i) \rangle$ in the *equation input* block, or as a lookup table, again in position $\langle \text{value}(i) \rangle$ in the *table input* block, where the first number in each table pair is temperature and the second is thermal conductivity.

$$[\text{rhocp} = \left\{ \begin{array}{l} \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{end}]$$

This statement is optional but it must be included if material properties within the segment are to vary with temperature. It defines the thermal capacity (product of material density and specific heat) in J/m³-K for the entire segment. There is no provision for this material property to vary spatially within an individual segment, except as temperature changes. The value is specified either as a constant with value $\langle \text{value}(r) \rangle$; as an equation, which may be a function of

“time” and/or “temp”, which appears as equation $\langle \text{value}(i) \rangle$ in the *equation input* block, or as a lookup table, again in position $\langle \text{value}(i) \rangle$ in the *table input* block, where the first number in each table pair is temperature and the second is thermal capacity.

$$[\text{hsrc} = \left\{ \begin{array}{l} \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{srcpf}, \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle, \text{end}]$$

This statement is optional but it must be included if material properties within the segment are to vary with time and/or temperature. It defines the local heat generation rate in units of W/m^3 for each of the node segments in the *thermseg/diffseg*. The specification consists of two parts. The first or temporal part identifies the magnitude and time history of the local heat generation rate. It may be specified as a constant with magnitude $\langle \text{value}(r) \rangle$; as an equation, which is a function of “time”, or as a lookup table, again in position $\langle \text{value}(i) \rangle$ in the *table input* block, where the first number in each table pair is time and the second is the local heating rate. The keyword **srcpf** identifies the data following it as the spatial distribution or source peaking factor profile. An entry must be made in this sequence for each node in the *thermseg/diffseg*. Those for the two surface nodes should be zero because they have no associated volume. A series of simple numerical values may be entered (one for each node or combined using the $\langle \text{numnds}(i) \rangle * \langle \text{value}(r) \rangle$ format).

The local heat generation rate is calculated as the product of the temporal part (applies to the entire segment) and the spatial part (an entry for each node). Hence, the magnitude of the heat generation rate may be associated with either part or divided between the parts, at the user’s discretion.

There are several options for boundary conditions (left and right) for heat transfer calculations.

$$\left[\left\{ \begin{array}{l} \text{htrbcl} \\ \text{htrbcr} \end{array} \right\} = \text{adiab}, \text{end} \right]$$

This option is for an insulated boundary through which there is no heat transfer.

$$\left[\left\{ \begin{array}{l} \text{htrbcl} \\ \text{htrbcr} \end{array} \right\} = \text{convect}, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\} \text{encl}, \langle \text{enclnbr}(i) \rangle, \text{end} \right]$$

This is the convective heat transfer boundary condition. Following the keyword **convect**, which identifies this as a convective boundary condition, the next entry identifies the specification for the convective heat transfer coefficient ($\text{W/m}^2\text{-K}$). It may be a real constant (with or without the keyword, **const**), an equation, or a table.

$$\left[\begin{Bmatrix} \text{htrbcl} \\ \text{htrbcr} \end{Bmatrix} = \text{stemp}, \begin{Bmatrix} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{Bmatrix}, \text{end} \right]$$

This is the specification for a fixed temperature boundary. That temperature may be specified as a real constant (with or without the keyword, **const**) or as an equation or a table in which the temperature is a function of time.

$$\left[\begin{Bmatrix} \text{htrbcl} \\ \text{htrbcr} \end{Bmatrix} = \text{sflux}, \begin{Bmatrix} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{Bmatrix}, \text{end} \right]$$

This statement specifies that the heat transfer boundary is of the fixed heat flux type. The heat flux is set as a real constant (with or without the keyword, **const**) or as an equation or a table in which the heat flux is a function of time. The algebraic sign assigned to the heat flux is directional. A positive value means the heat flux is to the right (increasing value of x). At a left face, that flux would be into the structure, while at a right face it would be out, and vice versa for a negative value.

$$\left[\begin{Bmatrix} \text{htrbcl} \\ \text{htrbcr} \end{Bmatrix} = \text{link}, \text{end} \right]$$

When structures are connected in series, the *link* boundary condition is used. This statement reflects that the current segment is linked to the one identified in the main input block. When the *link* boundary condition is employed with the connecting structure on the right of the current one (*htrbcr*), it is also required that the following statement appear.

$$[\text{hgap} = \begin{Bmatrix} < \text{value}(r) > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{Bmatrix}, \text{end}]$$

This statement defines the gap thermal conductance ($\text{W}/\text{m}^2\text{-K}$) between the two linked segments. It may be a real constant (with or without the keyword, **const**) or as an equation or a table in which the heat flux is a function of time. A high value indicates excellent thermal contact. This statement is only needed when the connecting segment is on the right of the current one.

In the following example, three segments are defined. Two are linked and have the full heat transfer calculations called for by the presence of the optional statements. The third is a very simple one with only one internal node and fixed temperature.

Example:

```
thermal input
  start thermseg,end
  delx=0.0,50*0.02,0.0,end
  tempd=52*300.0,end
  tcon=const,401.0,end
  rhocp=const,2439906.0,end
  hsrc=tabl,3,srcpf,norm,15.0,0.0467,0.125,398.0,end
  htrbcl=stemp,const,600.0,end
  htrbcr=link,end
  hgap=const,401.0,end
$
  start thermseg,end
  delx=0.0,50*0.02,0.0,end
  tempd=52*300.0,end
  tcon=const,80.2,end
  rhocp=const,3517890.0,end
  hsrc=const,0.0,srcpf,0.0,50*0.0,0.0,end
  htrbcl=link,end
  htrbcr=stemp,const,300.0,end
$
  start thermseg,end
  delx=0.0,1.0,0.0,end
  tempd=3*303.0,end

end of thermal input
```

4.4 Diffusion Input

start diffseg [,end]

This statement begins each *diffseg* definition sub-block. A new requirement is that the diffusion segment lattice number density must be provided for all options, so there is an added required statement:

nbrden = <value(r)>,end

Also, initial concentrations may now be given spatial distributions automatically:

concd =

$$\langle \text{dspc}(\alpha)_1 \rangle, \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle \\ \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle \\ \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle \end{array} \right\}$$

$$\begin{array}{l}
\left. \begin{array}{l}
< \mathbf{value}(r)_1 >, \dots, < \mathbf{value}(r)_n > \\
\mathbf{const}, < \mathbf{value}(r) > \\
< \mathbf{dspc}(\alpha)_2 >, \left\{ \begin{array}{l}
\mathbf{ramp}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 > \\
\mathbf{exp}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 >, < \mathbf{value}(r)_3 > \\
\mathbf{norm}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 >, < \mathbf{value}(r)_3 >, < \mathbf{value}(r)_4 >
\end{array} \right\}
\end{array} \right\} \\
\vdots \\
\left. \begin{array}{l}
< \mathbf{value}(r)_1 >, \dots, < \mathbf{value}(r)_n > \\
\mathbf{const}, < \mathbf{value}(r) > \\
< \mathbf{dspc}(\alpha)_m >, \left\{ \begin{array}{l}
\mathbf{ramp}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 > \\
\mathbf{exp}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 >, < \mathbf{value}(r)_3 > \\
\mathbf{norm}, < \mathbf{value}(r)_1 >, < \mathbf{value}(r)_2 >, < \mathbf{value}(r)_3 >, < \mathbf{value}(r)_4 >
\end{array} \right\}
\end{array} \right\} \\
\mathbf{end}
\end{array}$$

This statement establishes the initial concentration values of the mobile diffusion species defined in the *Main Input* block at each of the nodes if those values are other than 0. Values of diffusion species will be stored beginning at the left surface node of each segment and progressing to the right face. Values given in excess of the number needed for the number of nodes specified for the *diffseg* will be ignored. If an insufficient number of values is provided, it will be assumed that the remainder are zero. Note that the shorthand entry, $<\mathbf{numnds}(i)> * <\mathbf{value}(r)>$ may also be used to input multiple values of a given number. Functional input forms are implemented the same way as in the *temp* distributions described previously except for the *norm* distribution where the first parameter, $<\mathbf{value}(r)_1>$, is now the integrated number of atoms per square meter of surface area over the entire normal distribution (not counting the background value, normalized to lattice density) rather than the maximum value. If the distribution function is truncated by the surface, as it frequently will be, $<\mathbf{value}(r)_1>$ represents the number in the truncated distribution.

$$\begin{array}{l}
[\mathbf{ssconc} = <\mathbf{sspc}(\alpha)_1>, \left\{ \begin{array}{l} <\mathbf{value}(r) > \\ \mathbf{link} \end{array} \right\}, \left\{ \begin{array}{l} <\mathbf{value}(r) > \\ \mathbf{link} \end{array} \right\} \\
\\
[<\mathbf{sspc}(\alpha)_n>, \left\{ \begin{array}{l} <\mathbf{value}(r) > \\ \mathbf{link} \end{array} \right\}, \left\{ \begin{array}{l} <\mathbf{value}(r) > \\ \mathbf{link} \end{array} \right\}, \dots, \mathbf{end}]
\end{array}$$

This new optional statement is for specifying initial concentration values of any surface-only species identified in the *main input* block. Only those surface species identified here will be given non-zero initial values. The first entry following the surface species name is the initial value for the left face of the *diffseg*, while the second value is that for the right face. An entry of *link* for one face or the other should be entered if the *diffseg* is linked to another on that side. If

no surface species are to be identified for this *diffseg* or if both faces are linked, the statement may be omitted.

[trapping = ttyp,<ttyp(i)>

$$\text{tconc,} \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{ramp}, < \text{value}(r)_1 >, < \text{value}(r)_2 > \\ \text{exp}, < \text{value}(r)_1 >, < \text{value}(r)_2 >, < \text{value}(r)_3 > \\ \text{norm}, < \text{value}(r)_1 >, < \text{value}(r)_2 >, < \text{value}(r)_3 >, < \text{value}(r)_4 > \end{array} \right\}$$

$$\text{tspc}, < \text{dspc}(\alpha)_1 >, \text{alphr}, \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}$$

$$\text{alphr}, \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}$$

$$\text{ctrap,} \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{ramp}, < \text{value}(r)_1 >, < \text{value}(r)_2 > \\ \text{exp}, < \text{value}(r)_1 >, < \text{value}(r)_2 >, < \text{value}(r)_3 > \\ \text{norm}, < \text{value}(r)_1 >, < \text{value}(r)_2 >, < \text{value}(r)_3 >, < \text{value}(r)_4 > \end{array} \right\}$$

⋮

$$[\text{tspc}, < \text{dspc}(\alpha)_n > \text{alphr}, \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}$$

$$\text{alphr}, \left\{ \begin{array}{l} < \text{value}(r)_1 >, \dots, < \text{value}(r)_n > \\ \text{const}, < \text{value}(r) > \\ \text{equ}, < \text{value}(i) > \\ \text{tabl}, < \text{value}(i) > \end{array} \right\}$$

$$\begin{array}{l}
 \left. \begin{array}{l}
 \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\
 \text{const}, \langle \text{value}(r) \rangle \\
 \text{ctrap}, \left\{ \begin{array}{l}
 \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle \\
 \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle \\
 \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle
 \end{array} \right\}
 \end{array} \right\}] \\
 \vdots \\
 [\text{ttyp}, \dots] \\
 ,\text{end}]
 \end{array}$$

This is a new optional and somewhat complex statement that replaces all the trapping associated entries in TMAP4. It sets the number of different trap types that will be active in this *diffseg* together with all the other relevant data pertaining to these traps. These data include for each trap type (*ttyp*) the trap concentration (*tconc*), and for each trapped species (*tspc*) trapped in that trap type: the trap release rate (*alphr*) (1/s), which depends on trap type and is assumed to be species independent, the trapping rate coefficient (*alphr*) (1/s), which is species dependent but independent of trap type, and the concentration (*ctrap*) (fraction filled) of trapped atoms of that species initially in traps of that type. If trapping will be taking place in the problem, this statement must be included, at least one trap type must be specified, and at least one species must be trapped in each specified trap type. Not every *diffseg* need have trapping active, and if it is, only the trap types specified here will be assumed active in that segment. The maximum number of trap types in any *diffseg* is 3, but these need not be the same traps from one segment to the next. Only species identified in the main input block as diffusion species may be input following the *tspc* keyword.

$$\begin{array}{l}
 \text{dcoef} = \langle \text{spc}(\alpha)_1 \rangle, \left\{ \begin{array}{l}
 \langle \text{value}(r) \rangle \\
 \text{const}, \langle \text{value}(r) \rangle \\
 \text{equ}, \langle \text{value}(i) \rangle \\
 \text{tabl}, \langle \text{value}(i) \rangle
 \end{array} \right\}, \dots, \\
 \langle \text{spc}(\alpha)_n \rangle, \left\{ \begin{array}{l}
 \langle \text{value}(r) \rangle \\
 \text{const}, \langle \text{value}(r) \rangle \\
 \text{equ}, \langle \text{value}(i) \rangle \\
 \text{tabl}, \langle \text{value}(i) \rangle
 \end{array} \right\}, \text{end}
 \end{array}$$

This statement defines the diffusivity in the *diffseg* for each of the species in the problem. Entries are required for each surface-only species as well as for each diffusion species. Both are indicated here by the entry, $\langle \text{spc}(\alpha) \rangle$, and the order of entry is immaterial. In the case of surface-only species, the diffusivity is the lateral diffusivity. Like the standard diffusivity, this quantity may be entered as a constant, equation, or table.

$$\begin{aligned}
 \text{qstrdr} = & \langle \text{dspc}(\alpha)_1 \rangle, \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \dots, \\
 & \langle \text{dspc}(\alpha)_n \rangle, \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{end}
 \end{aligned}$$

This statement defines the quantity Q^*/k , or the Soret coefficient divided by Boltzmann's constant. Note that entries are needed for each diffusion species ($\langle \text{dspc}(\alpha) \rangle$) but not for surface-only species. Again, definition may be by constants, equations, or tables. The latter two may be functions of time or temperature.

$$\begin{aligned}
 \text{srcsd} = & \langle \text{dspc}(\alpha)_1 \rangle, \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\} \\
 & \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle \\ \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle \\ \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle \end{array} \right\} \\
 & \vdots \\
 & \langle \text{dspc}(\alpha)_n \rangle, \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\} \\
 & \left\{ \begin{array}{l} \langle \text{value}(r)_1 \rangle, \dots, \langle \text{value}(r)_n \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{ramp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle \\ \text{exp}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle \\ \text{norm}, \langle \text{value}(r)_1 \rangle, \langle \text{value}(r)_2 \rangle, \langle \text{value}(r)_3 \rangle, \langle \text{value}(r)_4 \rangle \end{array} \right\}, \text{end}
 \end{aligned}$$

This statement defines the volumetric source rate for the various diffusion species, such as from implantation or neutronic transmutation other than decay. Temporal dependence or flux intensity is defined by the expression following the diffusion species name. The statement has been changed to allow functional inputs to the source peaking factor, which provides spatial distributions. The initial keyword is *srcsd*. Note that because the species source rate is associated with volumetric production of a diffusion species, it should be specified as 0 for the two zero-volume surface nodes. For the functional forms, spatial distribution of diffusion species source rates will be accomplished automatically. Here in the *norm* distribution, the $\langle \text{value}(\mathbf{r})_1 \rangle$ parameter refers to the integrated value (atoms/m²s) over the thickness of the segment, not counting the background ($\langle \text{value}(\mathbf{r})_4 \rangle$) value. Hence, if all of the implanted flux intensity were to be stopped in a segment, the $\langle \text{value}(\mathbf{r})_1 \rangle$ value would be 1.0. The $\langle \text{value}(\mathbf{r})_2 \rangle$ parameter is the depth in meters of the midpoint of the distribution while the $\langle \text{value}(\mathbf{r})_3 \rangle$ parameter refers to the characteristic half-width of the distribution. For the other functional forms, the product of flux intensity and local peaking factor give the implantation flux in a given node. No species declared as surface-only species should be included in this statement.

```

{ diffbcl }
{ diffbcr } = surfdep,encl,<enclnbr(i)>

spc,<spc( $\alpha$ )1>,nu,<gnuz(r)>,ec<value(r)1>,[es,<value(r)2>]

[exch,<espc( $\alpha$ )1>,amu,<value(r)>,ex,<value(r)>]

[comb,<spc( $\alpha$ )>prob,<value(r)>]

[diss,<spc1( $\alpha$ )>,<spc2( $\alpha$ )>,eb,{
    <value(r)>
    const,<value(r)>
    equ,<value(i)>
    tabl,<value(i)>
}]

[form,<spc1( $\alpha$ )>,<spc2( $\alpha$ )>,prob,<value(r)>]

:

spc,<spc( $\alpha$ )n>,<gnuz(r)>,...

end

```

This new option for the diffusion boundary conditions indicates that surface energies control the flows of species to and from the surface and the formation and destruction of the species at the surface. The first entry is the user-assigned number of the enclosure, $\langle \text{enclnbr}(i) \rangle$, facing the surface.

A data sub-block should be entered for each species at the surface (surface-only or diffusion species) signified by the keyword *spc*. The species name is followed by the Debye frequency, $\langle gnuz(r) \rangle$, used in determining enclosure-exchange surface reaction rates. Normally that will be the Debye frequency of the material making up the surface, though it may be assigned in other ways. Next one enters the surface binding energy, E_c , denoted by the keyword *ec*, and entered in units of eV at $\langle value(r)_1 \rangle$. It may be followed by the solution enthalpy per atom, E_s , indicated by the keyword *es*. It is entered in units of eV at $\langle value(r)_2 \rangle$. These are used in establishing asymmetric diffusion from the surface to the subsurface sites. If the current surface species is a surface-only species, the E_s statement should be omitted because, by definition, surface-only species do not exist in the bulk. Following that are four different reaction types that may (but each need not) be specified. These are (1) *exch* or capture and escape of the surface species to the enclosure, (2) *comb* or the combination of the surface species with another species at the surface to form a different one, (3) *diss* or the dissociation of the present surface species into two different ones, and (4) *form* or the formation of the present surface species from two different ones. Any or all of these four sub-statements should be repeated for each reaction that involves the present surface species. These will now be discussed briefly.

The optional (under *surfdep* conditions) *exch* option, or exchange with the enclosure, is a thermally activated process in which the surface species overcomes its binding potential and escapes in its current molecular state to the enclosure gas. Likewise, molecules in the enclosure overcome the barrier potential and attach to the surface. Required data include

- The identity of the enclosure species, $\langle espc(\alpha)_1 \rangle$, by which the escaping diffusion or surface-only species will be known in the enclosure domain (note that this may be the same name, but it is a separate entity from the diffusion or surface-only species name).
- The molecular mass in amu of the enclosure species following the *amu* keyword.
- The capture barrier energy, indicated by the keyword *ex*, of the surface species, entered in units of eV at $\langle value(r)_1 \rangle$ (corresponds with E_h in Figure 1). Note that this value should not be less than zero.

The second or *comb* option is a combination of the present surface species with another to form a third species, denoted by the keyword *comb*. This is a loss mechanism for the given species at the surface. The first required datum for this specification is the identity of the surface species, $\langle sspc(\alpha) \rangle$, with which the present species combines to form the new species. Because each reaction is defined and computed in terms of the present surface species, the identity of the species formed is not necessary at this point. However, this same reaction should appear as a formation reaction for that different species. Activation frequencies for the combination reaction are taken from the diffusivities entered in the *dcoef* statement. The second datum is the probability that the combination reaction will take place given that the combining species find each other. It is entered following the keyword *prob*. The value of this probability may not exceed unity.

The third option for the surface reactions is the dissociation reaction, indicated by the keyword *diss*. This is another loss mechanism for the surface species. Data required here include

the activation energy for the dissociation, entered in units of eV at $\langle \text{value}(r) \rangle$ and, after the keyword **prod**, the identities of the product species formed. Note that the products of this reaction should *not* be included in the **form** group since the product is formed from a single species, not as a combination. TMAP7 keeps track of the species formed by dissociation automatically.

The final option is the formation of the surface species from two different surface species, indicated by the keyword **form**. This is a source term for the surface species. Required data here include the identities of the two surface species that combine to form the present one. These names are entered at $\langle \text{sspc}_1(\alpha) \rangle$ and $\langle \text{sspc}_2(\alpha) \rangle$. Activation frequencies for the combination reaction are taken from the diffusivities entered in the *dcoef* statement. The final datum for this reaction is the probability that the combination reaction will take place given that the species forming the current surface species find each other. It is entered following the keyword **prob**. The value of this probability may not exceed unity.

Under the *form* specification, there must be a **spc** entry for each species at the surface (maximum 40; 10 diffusion and 30 surface-only). Within each **spc** block, there must be a data set for each molecular transformation that takes place involving that surface species (max 40).

$$\left\{ \begin{array}{l} \text{difbcl} \\ \text{difbcr} \end{array} \right\} = \text{ratedep, encl, } \langle \text{enclnbr}(i) \rangle$$

spc, $\langle \text{sspc}(\alpha)_1 \rangle$

$$\text{exch}, \langle \text{espc}(\alpha) \rangle, \text{ksubd}, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}$$

$$\langle \text{spc}(\alpha) \rangle, \text{ksubr}, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}$$

[comb, $\langle \text{sspc}(\alpha) \rangle$, prob, $\langle \text{value}(r) \rangle$]

[form, $\langle \text{sspc}_1(\alpha) \rangle$, $\langle \text{sspc}_2(\alpha) \rangle$, prob, $\langle \text{value}(r) \rangle$]

$$[\text{diss}, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{prod}, \langle \text{sspc}_1(\alpha) \rangle, \langle \text{sspc}_2(\alpha) \rangle]$$

⋮

spc,<spc(α)_n>,. . .

end

This diffusion boundary condition block is very similar to that for the *surfdep* option. Several key differences should be pointed out. First, the *exch* sub-block is not optional. It is assumed that when the *ratedep* option for diffusion boundary conditions is selected, there will be conventional recombination and dissociation at the surface. However, as with the *surfdep* option, there may be surface conversion reactions that do not involve exchange with the surrounding gas going on as well. As with the *surfdep* case, it is necessary to enter a block for each surface-only and diffusion species defined in the problem, and a sub-block for each reaction involving that species. Next, instead of these reactions proceeding at rates characterized by the Debye frequency and some activation energies, the exchange recombination and dissociation processes must be given specifications in terms of constants, equations, or tables. Normally the equations and tables will be functions of temperature. For the *comb* and *form* options, rates are taken from diffusivities entered in the *dcoef* statement.

$\left\{ \begin{array}{l} \text{difbcl} \\ \text{difbcr} \end{array} \right\} = \text{lawdep, encl, <enclnbr(i)>}$

dspc,<dspc(α)₁>,<espc(α)>,pexp,<value(r)>,solcon, $\left\{ \begin{array}{l} < \text{value(r)} > \\ \text{const, < value(r) >} \\ \text{equ, < value(i) >} \\ \text{tabl, < value(i) >} \end{array} \right\}$

:

dspc,<dspc(α)₁>,<espc(α)>,pexp,<value(r)>,solcon, $\left\{ \begin{array}{l} < \text{value(r)} > \\ \text{const, < value(r) >} \\ \text{equ, < value(i) >} \\ \text{tabl, < value(i) >} \end{array} \right\}$

end

This statement defines an equilibrium or *lawdep* boundary condition. The first entry is the enclosure number into which gas exchange is occurring. Following that, an entry block is made for each diffusion species. Surface-only species are not defined for *lawdep* boundary conditions because the solution laws only apply to diffusing species. After giving the diffusing species name, the homonuclear enclosure species formed in the reaction (or corresponding identity enclosure species if Henry's law applies) with the current diffusion species is identified. Note that it is possible to treat just one part of a heteronuclear molecule as a Henry's law reaction when there is no homonuclear reaction available. Then, following the keyword *pexp*, the exponent from the solution law

$$C_{<sspc(\alpha)>} = solcon \times P_{<espc(\alpha)>}^{pexp} \quad (62)$$

is provided followed by the keyword **solcon**, and the solubility for the homonuclear (or identity) molecule of that species in the material. The exponent **pexp** is normally 0.5 for Sieverts' law solubility and 1.0 for Henry's law solubility. This solubility rule is used to establish surface concentrations of the atomic species corresponding to the specified homonuclear (or identity) molecules under the *lawdep* boundary condition. A **dspc** block must be entered for each diffusion species in the problem. Surface-only species are not entered because their reactions will have no effect on the surface concentrations of diffusion species.

$$\left\{ \begin{array}{l} \text{difbcl} \\ \text{difbcr} \end{array} \right\} = \text{sconc}$$

$$\text{dspc}, <\text{dspc}(\alpha)_1>, \text{conc}, \left\{ \begin{array}{l} <\text{value}(r)> \\ \text{const}, <\text{value}(r)> \\ \text{equ}, <\text{value}(i)> \\ \text{tabl}, <\text{value}(i)> \end{array} \right\}$$

$$\vdots$$

$$\text{dspc}, <\text{dspc}(\alpha)_n>, \text{conc}, \left\{ \begin{array}{l} <\text{value}(r)> \\ \text{const}, <\text{value}(r)> \\ \text{equ}, <\text{value}(i)> \\ \text{tabl}, <\text{value}(i)> \end{array} \right\}, \text{end}$$

This statement defines fixed concentrations of diffusion species at the surface of a *diffseg*. After the defining keyword **sconc**, there must be a **dspc** entry block for each of the diffusion species defined in the problem, but no surface-only species are entered. Following the surface species name is the surface concentration specification for that species. That may be in constant, equation, or tabular form. Table look-ups assume time is the independent variable in the relation defined by the table while time or temperature may be independent in equations. A difference from the previous version is that neither the enclosure number nor the stoichiometry are needed because the **sconc** boundary condition is for setting the surface concentration of a diffusion species. Because it is strictly a fictitious boundary condition, provided only to examine certain hypothetical situations, no molecular flows to the surrounding enclosures are calculated. However, diffusive fluxes of diffusion species are determined and, if requested, recorded in the *pltdata* file.

$$\left\{ \begin{array}{l} \text{difbcl} \\ \text{difbcr} \end{array} \right\} = \text{link}, <\text{dspc}(\alpha)_1>\text{solcon}, \left\{ \begin{array}{l} <\text{value}(r)> \\ \text{const}, <\text{value}(r)> \\ \text{equ}, <\text{value}(i)> \\ \text{tabl}, <\text{value}(i)> \end{array} \right\}, \dots,$$

$$\langle \text{dspc}(\alpha)_n \rangle, \text{solcon}, \left\{ \begin{array}{l} \langle \text{value}(r) \rangle \\ \text{const}, \langle \text{value}(r) \rangle \\ \text{equ}, \langle \text{value}(i) \rangle \\ \text{tabl}, \langle \text{value}(i) \rangle \end{array} \right\}, \text{end}$$

This statement defines the left or right edge of this *diffseg* as being linked to one adjacent to it. For each diffusion species, there must be a solubility constant defined following the keyword, *solcon*.

$$\left\{ \begin{array}{l} \text{difbcl} \\ \text{difbcr} \end{array} \right\} = \text{nonflow}, \text{end}$$

This statement indicates that no diffusive flows cross this surface of the *diffseg*.

surfa = <value(r)>,end

This statement defines the cross sectional area of the *diffseg*, assumed to be uniform over the length of the *diffseg* which was defined in the *thermal input* block. Units of area are square meters.

Example:

```
start diffseg,end
$ 15-nm implantation zone [DIFFSEG 1]
nbrden=6.25e28,end
concd=h,const,1.0e-10,end          $ Starting mobile concentration
ssconc=h2,1.0,link,end            $ Starting surface species concentration
trapping=ttyp,1,tconc,norm,0.13,4.6e-9,1.0e-8,0.0,tspc,h,alphr,equ,4
    alpht,equ,3,ctrap,const,0.0
    ttyp,2,tconc,const,3.2e-2,tspc,h,alphr,equ,5
    alpht,equ,3,ctrap,const,4.4e-10
    ttyp,3,tconc,const,1.0e-3,tspc,h,alphr,equ,6
    alpht,equ,3,ctrap,const,1.4e-10,end
qstrdr=h,const,0.,end              $ Q*/R for Soret effect unknown
dcoef=h,equ,7,h2,equ,10,end        $ Diffusion coeff (m2/s) [Modified]
srcsd=h,tabl,2,srcpf,norm,1.0,4.6e-9,3.0e-9,0.0,end
difbcl=surfdep,encl,1
    spc,h,nu,8.4e12,ec,-0.8,es,1.04
    comb,h,prob,1.0
    spc,h2,nu,8.4e12,ec,-0.1
    exch,h2g,amu,2.0,ex,0.05
    diss,h,h,eb,0.05
    form,h,h,prob,1.0,end
difbcr=link,h,solcon,equ,8,end
surfa=0.0025,end                  $ 50 x 50 mm square
$
start diffseg,end
$ Balance of 0.1-mm tungsten specimen [DIFFSEG 2]
nbrden=6.25e28,end
concd=h,const,1.0e-10,end          $ Starting mobile concentration
ssconc=h2,link,1.0,end            $ Starting surface species concentration
trapping=ttyp,1,tconc,const,0.0,tspc,h,alphr,equ,4
```

```

        alpht,equ,3,ctrap,const,0.0
        ttyp,2,tconc,const,3.2e-2,tspc,h,alphr,equ,5
        alpht,equ,3,ctrap,const,4.4e-10
        ttyp,3,tconc,const,1.0e-3,tspc,h,alphr,equ,6
        alpht,equ,3,ctrap,const,1.4e-10,end
qstrdr=h,const,0.,end          $ Q*/R for Soret effect unknown
dcoef=h,equ,9,h2,equ,10,end    $ Diffusion coeff (m2/s) [Modified]
srcsd=h,const,0.0,srcpf,18*0.0,end
difbcr=surfdep,encl,1
    spc,h,nu,8.4e12,ec,-0.8,es,1.04
    comb,h,prob,1.0
    spc,h2,nu,8.4e12,ec,-0.1
    exch,h2g,amu,2.0,ex,0.05
    diss,h,h,eb,0.05
    form,h,h,prob,1.0,end
difbcl=link,h,solcon,equ,8,end
surfa=0.0025,end              $ 50 x 50 mm square
$
end of diffusion input

```

4.5 Equation Input

Equations are entered in the *Equation Input* section. The section begins with the statement

equation input

Between the beginning and closing statements for this block, equations are input in the general form

y = f(time,temp,concd,conce),end

Any legitimate FORTRAN expression can be defined as an equation, including functions with more than one argument separated by commas. Each equation must end with “**end**”. Comment statements beginning with “\$” may be used between equations or after the “**end**” of each equation, but not between continuing lines of the equation definition. The time parameter is invoked with the variable “time”. Temperature is invoked with “temp”. The context of the function for which the equation is defined will tell TMAP7 whether it is a diffusion node temperature or an enclosure temperature that is called for. Concentrations called for need to be specified as either “concd” for concentrations of diffusing species or “conce” for enclosure species. Additionally, the index of the particular concentration in the *concd* or *conce* arrays must be provided. Indexing is linear. For the diffusion species, the count starts in *diffseg* 1 with species 1 at node 1 and proceeds first through the nodes in that segment, next through the species present in the segment, and finally through the sequence of segments. Surface-only species are accessed through “concd”, indexed beginning at a number one higher than all the diffusion species in all the nodes and proceeding in pairs by species number and then by structure, the first for the concentration on the left face of the structure, and the second for the concentration on the right face. For enclosure species, indexing begins at the first enclosure defined for the first species defined and proceeds through the diffusion species in the enclosure and then through the enclosures in the sequence in which they were defined. The section closes with

end of equation input

Example:

```
equation input
$ (1) Thermal conductivity of tungsten (W/m-K)
y=163.-0.0739*temp+2.89e-5*temp**2-4.3e-9*temp**3,end
$ (2) Rho Cp for tungsten (J/m3K)
y=(1930.-.0388*temp)*(131.+0.0226*temp-5.73e-6*temp**2+3.69e-9
*temp**3),end
$ (3) Alpkt for h in tungsten (1/s)
y=9.1316e12*exp(-0.39/8.625e-5/temp),end
$ (4) Alplr for trap 1 in tungsten (1/s)
y=8.4e12*exp(-1.3/8.625e-5/temp),end
$ (5) Alplr for trap 2 in tungsten (1/s)
y=8.4e12*exp(-1.75/8.625e-5/temp),end
$ (6) Alplr for trap 3 in tungsten (1/s)
y=8.4e12*exp(-3.1/8.625e-5/temp),end
$ (7) Diffusivity for h in tungsten (m2/s)
y=4.1e-7*exp(-0.39/8.625e-5/temp),end
$ (8) Hydrogen solubility in tungsten (1/m3-Pa^1/2)
y=1.83e24*exp(-1.04/8.625e-5/temp),end
$ (9) Diffusivity for h in implant-layer tungsten (m2/s) [Modified]
y=4.1e-7*exp(-.39/8.625e-5/temp)-1.e-10*concd(21)*exp(-0.005*time),end
$ (10) Surface diffusivity for h2 at tungsten surface (m2/s)
y=4.1e-7*exp(-.1/8.625e-5/temp),end
end of equation input
```

4.6 Table Input

Tables are input in the *table input* section, which begins with the statement

table input

Tables are entered as real number pairs, each number separated by a comma from the one following except a number at the end of an input line, where the carriage return at the end of the line serves as the delimiter. It is mandatory that each independent variable value (numbers 1,3,5,...) in each table be larger than the one before it. Present and somewhat arbitrary dimensioning limits require that the total number of tables not exceed 100 and that the total number of number pairs be no greater than 550 for all tables (1,100 numbers in all tables combined). Each table must end with the “**end**” keyword. Comments may appear between tables, but they may not interrupt a table listing. The *table input* block ends with the statement

end of table input

Example:

```
table input
$ (1) Temperature history
0.0,825.,8.e5,825.,end
$ (2) Pressure history of D2 in Enclosure 1
0.,1.e-4,300.,1.e-4,301.,0.02,600.,0.02,601.,0.04,900.,0.04,901.,0.06
```

```
1200.,0.06,1201.,0.08,1500.,0.08,1501.,0.1,8.e5,0.1,end
end of table input
```

4.7 Control Input

The control input section controls the basic solution management parameters for the problem. It begins with the statement

control input

The order in which the *control input* parameters are entered is not significant, but most statements are required. In some cases, default values will be inserted if no values are entered.

time = <tstrt(r)>,end

This statement initializes the time variable. Typically the value will be 0.0, but it may be set at any arbitrary value. Time is measured from that point.

tstep = <tstp(r)>,end

Here, the time step to be used for forward marching computations is set. The value entered in the *tstep* statement is the maximum size time step used in the problem. However, during operation, if the number of iterative cycles, either in the diffusion loop or in the enclosure loop, exceeds the maximum number allowed by the *itermx* statement specification, the time step is reduced by a factor of 10 to improve convergence. The process repeats if needed until the factor reaches 10^{-7} times the original time step size. Then, the problem will stop for lack of convergence. Each 100 cycles, the code increases the time step by a factor of 10 and tries for convergence. If unsuccessful, the time step goes back down. Otherwise, it increases by stages up to its original value. Reduced time steps do not change the interval at which print output edits or PLTDATA values are recorded.

timend = <tmnd(r)>,end

This statement sets the time at which computations end. It must be greater than the starting time entered in the *time* statement. Note that the end time is not elapsed time after the problem start, but when the *time* value in the problem exceeds *timend*, the problem stops. Thus it must be greater than the starting time discussed in the earlier statement.

nprint = <ncyc(i)>,end

This statement defines the number of *tstep* cycles skipped between output listings of the computational results. The print will occur when $\text{MOD}(\text{time}/(<\text{ncyc}(i)>*\text{tstep})) = 0.0$.

itermx = <mcyc(i)>,end

Many problems require iterative solutions that make use of iterative convergence. This

statement defines the number of iterations that can be performed before the program stops or takes other actions. If this statement is not supplied, the code will use the value `<mcyc(i)> = 1,000`.

delcmx=<eps(r)>,end

This statement defines the convergence limit, treated in most cases as a value fraction, beyond which the solution is deemed to converge. If the value is very small, the code treats it as the magnitude of the correction itself that will constitute convergence. If no value is supplied, the code uses `<eps(r)> = 1.0E-07`.

Changes from TMAP4 incorporated into this section include addition of the following statements:

bump=<bump(r)>,end

This statement defines the fractional change in dependent variable value used in the process of Newton-Raphson convergence. If no value is supplied, TMAP7 assumes a value of `1.0E-02`.

bound=<bound(r)>,end

This statement sets the limit of fractional change allowed in updating Newton-Raphson convergence independent variables. This is intended to prevent numerical run-away when derivatives are relatively small. If the solver calls for an increase in the size of the iteration variable, the increase is limited to the factor *bound* times the present value of that variable. If the solver wants to decrease the size of the iteration variable, the change is limited to *bound*/(1 + *bound*) times the current value. When no value is provided, the code assumes a value of `2.0`.

omega=<omega(r)>,end

This is the standard over-relaxation or damping factor used to accelerate or retard convergence in the diffusion part of the solution. Typically a value of `1.3` is good, but different values may prove better on particular problems. The `1.3` value is assumed if none is provided.

damp=<damp(r)>,end

This operates like *omega* above, but it is a damping factor used to accelerate or retard convergence in the enclosure part of the solution. Typically a value of `0.7` is good, but different values may prove better on particular problems. The `0.7` value is assumed if none is provided.

Example:

```
control input
  time=0.0,end
  tstep=0.01,end
  timend=1800.0,end
  nprint=10000,end
  itermx=900,end
  delcmx=1.0e-5,end
  bump=1.e-3,end
```



```
damp=0.7
bound=4.0,end
omega=1.3,end
end of control input
```

4.8 Plot Input

The *plot input* block is used to control data being printed to the PLTDATA file. This file is used to store tabular output for later plotting, and the data are accessed using the EXTRACT utility described later. The block begins with the statement

plot input

Then come the various statements to define data to be included in the PLTDATA file:

nplot = <value(i)>,end

This statement defines the frequency at which plot data are recorded in the PLTDATA file. An integer n entered means data will be recorded after every n th computational time step.

plotseg = [<segno(i)>,<segno(i)>,...,]end

This statement defines which of the diffusion segments will be included in the PLTDATA file listing. Even if the problem has only one diffusion segment, the statement must be included. Segments are numbered in the order they were defined in the *main input* section. Omission of a *segno* list means no segment data will be recorded.

plotencl= [<enclno(i)>,<enclno(i)>,...,]end

This statement defines which of the enclosures will be included in the PLTDATA file listing and is a required statement. Enclosures are numbered using the numbers assigned by the user in the *enclosure input* data block. Omission of an *enclno* list means no enclosure data will be recorded.

dname = [<spcnm(α)>,<spcnm(α)>,...,]end

This statement determines which of the diffusion species will be included in the PLTDATA file listing. Species names are identified using the names given in the *main input* data block. If no names are entered, no diffusion species data will be included. This list does not include surface-only species.

[sname = [<sspc(α)>,...,<sspc(α)>],end]

This statement identifies which surface-only species are to be included in the PLTDATA file. Because the presence of surface-only species is optional, this entire statement is optional. Omission of the statement or omission of the *sspc* data list will result in no surface-only species data being included. Names used are those given in the *main input* data block.

ename = [<spcnm(α)>,<spcnm(α)>,...,]end

This statement determines which of the enclosure species will be included in the PLTDATA file listing. Species names are identified using the names given in the *main input* data block. If no names are entered, no enclosure species data will be included.

dplot = [stemp,] [sconc,] [moblinv,] [sflux,] [trapinv,] end

This statement determines the type of diffusion-species relevant data to be recorded. Each of the option parameters is optional. While the *dplot* statement must be included, omission of option parameters means no data will be recorded. All parameters must be entered using the exact keyword given.

The *stemp* parameter causes diffusion segment surface temperatures to be included for each of the diffusion segments identified in the *plotseg* statement above.

The *sconc* parameter causes surface concentration values to be included for each of the diffusion species identified in the *dname* statement and for each surface-only species included in the *sname* statement, for each diffusion segment included in the *plotseg* statement.

The *moblinv* parameter causes mobile inventory values to be listed for each of the diffusion species identified in the *plotseg* statement, in each of the diffusion segments identified in the *dplot* statement.

The *sflux* parameter causes values to be included for diffusion flux at both surfaces of each diffusion segment identified in the *dplot* statement for each species identified in the *dname* statement. Fluxes are directional with positive values being in the left to right (increasing x) direction.

The *trapinv* parameter includes inventories of each species called for in the *dname* statement in each of the traps identified in the *diffusion input* block and for each diffusion segment identified in the *plotseg* statement.

eplot = [press,] [conv,] [diff,] [etemp,] end

This statement defines which enclosure data will be included in the PLTDATA file. Each option may be included or omitted at the user's discretion. Omitting them all will result in no enclosure data being included.

The *press* parameter includes species pressures for each enclosure species identified in the *ename* statement and for each enclosure identified in the *plotencl* statement.

The *conv* parameter includes integrated convective flows into the enclosure of all species identified in the *ename* statement, for each enclosure identified in the *plotencl* statement. Positive values are into the enclosure from other enclosures.

The *diff* parameter includes the integrated diffusive flows into the enclosure for each species identified in the *ename* statement.

The *etemp* parameter includes the history of enclosure temperature.

Example:

```
plot input
  nplot=500,end
  plotseg=1,end
  plotenc1=2,3,5,end
  dname=h,d,end
  ename=h2,hd,d2,end
  dplot=moblinv,sflux,end
  eplot=press,conv,diff,end
end of plot input
```

4.9 Terminal Declaration

The input file is closed with the terminal declaration

end of data

5.0 CODE USAGE

In this chapter, we present the requirements for installation and use of TMAP7. The code was developed specifically for running on a PC under the WINDOWS™ operating environment. It actually is run from the “Command Prompt” or “DOS” window. However, because it is based on the public license GNU FORTRAN, which operates on a variety of operating platforms, it should run well in non-windows environments as well. However, at this writing, other environments have not been verified.

5.1 Installation

5.1.1 General

We assume in what follows that the user has a working knowledge (need not be expert) of the computer system on which TMAP7 is to be used and of its operating system (WINDOWS XP for example). TMAP7 is furnished as a self-extracting archive on a CD or as a download that contains all files necessary to install and run TMAP7. These include the GNU public license FORTRAN-77 compiler with its supporting files, FORTRAN source files for the preprocessor, computational, and data extraction modules, compiled (.O) and executable (.e) files for these modules, DOS batch files, and ASCII files of a number of sample programs and helpful information. The files in the main folder and their purposes are as follows:

EQU.O	A dummy linkable equation object file. It is linked if no equations are used in the input file and no TAPE7.OBJ module is produced by the preprocessor.
EXTRACT.EXE	An executable file for the code that generates x-y pairs for plotting from the PLTDATA file.
EXTRACT.FOR	The FORTRAN-77 source code for EXTRACT.EXE .
G.BAT	A necessary utility for temporarily adjusting the PATH statement in the AUTOEXEC.BAT file so G77 can find the library functions it needs. This file is listed in the Appendix.
G77READ.TXT	An introduction to the GNU G77 compiler furnished by its developers.
READ.ME	An ASCII text file with installation instructions and supplementary data.
T7.BAT	A DOS batch file to compile and run a problem using TMAP7. It automatically adds suffixes of .INP for the input file, .OUT for the output file, and .PLT for the PLTDATA file (used to generate x-y pairs for plotting). A listing of T7.BAT is provided in the Appendix for reference.
TMAPC7.FOR	FORTTRAN-77 source code for the main computational program module. This is the part of the code that performs the calculations.

TMAPC7.O	The compiled version of TMAPC7.FOR that is linked to form the executable code that performs the calculations. Do not link this module. Linking is done by T7.BAT during execution.
TMAPP7.EXE	The executable pre-processor, used manually or invoked by the T7.BAT
TMAPP7.FOR	The FORTRAN-77 source code for TMAPP7.EXE .
TMAP7Man.PDF	This document
V&V.PDF	A .PDF version of the verification and validation report, Ref. 1.

In addition, there are three folders holding the GNU FORTRAN compiler and its associated files: *bin*, *doc*, and *lib*. These hold, respectively, the binary files of the GNU G77 compiler, the documentation files on this compiler in HTML format, and the library of functions and other operations to be called during the execution. A fourth folder, *samples*, holds .INP, .OUT, and .PLT files for the problems included in the verification and validation report.¹

The general concept of TMAP7 operation is that the user generates an input data file that contains the specification of the problem (the topic of Chapter 4 of this manual). Computations are initiated by entering the batch file name, **T7**, followed by the root of the problem input file name. For example, suppose the input file name was MYPROB.INP (it is important to have the .INP name extension if the automated operation is to be successful). The user would enter **t7 myprob** to start the problem. The input file (*.INP) is first copied by **T7.BAT** into a file called **TMAPINP**, which is called for by the preprocessor (**TMAPP7**), leaving the original input file untouched for future reference. If in other than a DOS (WINDOWS™ command prompt) environment, the functions of the **T7.BAT** file will need to be accomplished manually or by an equivalent script file. The **TMAPINP** file is processed by the preprocessor (**TMAPP7**) into an object file (called **TAPE1.O**) that is read by the computational module (**TMAPC7.O**) during its operation to form an executable binary file (**A.EXE**).

If equations are included in the *equation input* block of the input file, such as for specifying transport parameters or state-variable histories, the preprocessor generates a callable subroutine module called **TAPE7.FOR** that is subsequently compiled into the object file **TAPE7.O** that contains the compiled equations. If no equations are used the input file, then the FORTRAN-77 compiler is not needed, and the dummy file, **EQU.O**, or a surviving **TAPE7.O** file generated in a previous run, is linked in place of **TAPE7.O**. If there is a question as to how the equations were translated into FORTRAN, the **TAPE7.FOR** file can be edited.

Execution takes place, producing two ASCII output files and an ASCII summary data file. The first of the output files, **PREPOUT**, is generated by the preprocessor. It contains the listing of the input file and any diagnostic comments generated. If errors are detected by the preprocessor, the computational part of the code does not execute. The second output file, **CODEOUT**, is only generated if the computational code actually executes. It contains the information produced to document the intermediate results of the calculations at intervals of each user-specified time step. The **PREPOUT** and **CODEOUT** files are combined by **T7.BAT** into

an ASCII output file (user-specified root name with the **.OUT** extension). Again, if in other than a DOS environment, this combining of files, if desired, will need to be accomplished manually or by another script file. The tabular plot data or **PLTDATA** file (copied to a file with the user-specified root name with the **.PLT** extension) contains tabulated data from which x-y data pairs for plotting may be extracted using **EXTRACT.EXE**. Plotting may be done with a user-supplied program such as a spreadsheet or other graphics package. The **TMAPINP**, **TAPE1**, **TAPE7.O**, **PREPOUT**, and **CODEOUT** files remain after computations are complete and **PREPOUT** and **CODEOUT** have been copied to the *.OUT file and **PLTDATA** has been copied to the *.PLT file. They are deleted during initialization for the next computation. The linked executable file, **A.EXE**, is deleted at the end of the computations.

Because the **TAPE1** file is written as a binary file, quantities passed from the preprocessor to the computation module are not directly visible. Many of those values are written in ASCII format in a file called **MEMOUT**, which may be of some use to the experienced TMAP7 user.

The code calculates real numerical values in double precision arithmetic. Integers are generally in integer*2 format but those used as indexes are in integer*4, consistent with rules for the GNU FORTRAN compiler.

5.1.2 Setup

TMAP7 comes as a self-extracting archive or zipped download file called **TMSETUP.EXE**. It uses automatic file copying to install the code in a folder of the user's choice. The default folder is C:\TMAP7. During extraction, the user is given the choice to save it elsewhere. Locate and run the file called **TMSETUP.EXE**.

5.2 Operation

5.2.1 TMAP7 Code

To run TMAP7, open the WINDOWS™ command prompt window and go to the directory in which TMAP7 is installed. Run the **G.BAT** file by entering the letter [g], then [Enter]. This updates the environment so the GNU G77 compiler can find its library files.

Edit input files using the **EDIT** line editor provided with WINDOWS or some other text editor. Files must be simple ASCII text. Input files must end with the filename extension **.INP**.

To run the TMAP7 code, simply type **t7 <filename root>** for example **t7 myprob**. If the **.INP** file is in a different directory, include the path to the file in the **<filename.root>** entry. For example, suppose **myprob.inp** was in a subfolder to the TMAP7 folder called **work**. The command to execute TMAP7 using that file would be

t7 work\myprob

If the input file were **expermt1.inp** on a jump drive assigned drive letter E:, the command would be

t7 e:\work\expermt1

Output files (.OUT and .PLT) will be written to that same location

If the processing is successful, the screen will flash the message “TMAP7 PROBLEM IN PROGRESS”, and the display will indicate progress through the problem with a sequence of status writes. On the left after the keyword **ncyc=** is the current iteration number. Next is the message **time(s)=** followed by the elapsed problem time. Then, **dloop=** and **eloop=** indicate the number of iterations required for diffusion species computations and enclosure species computations. The **last=** number is the currently estimated number of iteration cycles remaining to complete the problem. Asterisks in this field mean the number is larger than can be displayed with 10 characters. At the end of the problem, the output file will be tagged with

***** NORMAL PROBLEM TERMINATION REACHED *****

and the screen will display the message

*****PROBLEM EXECUTION COMPLETE*****

If there has been a problem, there will be no such indication. Check diagnostic messages from the system that will appear on the screen. Edit the **PREPOUT** file to see messages from TMAP7 on file construction or other problems that may have arisen. Diagnostics from TMAP7 in the **PREPOUT** file may be somewhat illusory. Begin correcting the input file at the first error indication, and later error messages may vanish.

If a message appears indicating the problem has been stopped, the reason is probably associated with numerical convergence. Try reducing the **omega** and/or **damp** parameters in the *control input* block of the input file.

To stop execution of the problem, enter [Ctrl]+[c]. Data written to the CODEOUT file to that point will be saved.

5.2.2 Extract Utility

The **EXTRACT.EXE** utility generates lists of number pairs, one list at a time, for plotting. Each list is has problem time as the first number of each pair with a value selected by the user as the second number in the pair. The utility is initiated by entering the program name, for example

C:TMAP7>extract

where the first part of the line shown is the prompt given by WINDOWSTM. The program will then request a name for the file it is to operate on. If the TMAPC7 code has just been run, one may respond with **PLTDATA**. That is the default file name, and simply pressing [Enter] will select that file. Any **PLTDATA** file name such as **MYPROB.PLT** may also be given. After entering that file name, EXTRACT will ask for a file name for its own output. It may be convenient to give that file the same root name as the problem that generated the data followed by an extension indicative of its contents (e.g., **MYPROB.PH1**). EXTRACT will then present

the user with a number of choices to select the time-dependent data for this plot file, depending on the *plot input* data block parameters. When the choices are complete, the file will be written, and a message will confirm the filename written to.

5.3 Restart Capability

TMAP7 includes a restart capability to allow for alterations of equations, tables, or any *control input* parameter. The TMAPC7 code module will create a *restart* file called **RSTOUT** at the completion of the problem, when and if the specified problem end time has been reached. This file, if present, is overwritten during the execution of TMAP7, so if later restarting is contemplated, the **RSTOUT** file must be saved under a different name before running TMAP7 for another problem.

Restarting a problem involves several steps:

1. Ensure the presence of the previously generated **RSTOUT** file. Change the local file name to **RSTIN** (for example enter **rename rstout rstin**).
2. Revise the input file (see Chapter 4 for input file definition) as follows:
 - a. The first non-comment card of the input deck should be **restart**.
 - b. Remove the *title input*, *main input*, *enclosure input*, *thermal input*, *diffusion input*, and *plot input* data blocks, leaving only *equation input*, *table input*, and *control input* data blocks and the **end of data** declaration.
 - c. Be sure that the number and sequence of equations in the *equation input* block and the number and sequence of tables in the *table input* block remain the same in the restart file as in the original input file. The content of equations and tables may change.
 - d. Use of the **time** statement in the *control input* block is optional when restarting the problem. If included, the new value of starting time will override the elapsed time calculated in the previous execution. If left out, the starting time for the restart execution will be the ending time of the previous execution. All other *control input* parameter statements are required. Note that the **timend** value is measured with respect to the total problem time and must exceed the new starting time if any calculations are to be accomplished.
3. Provide the restart ***.INP** file with a modified root name to prevent overwriting of the already written ***.OUT** and ***.PLT** files by TMAP7 during the continued execution. For example, if the original input file was named **PROB.INP**, the user may wish to name the restart input file **PROB-1.INP**. Note that to operate under the DOS batch executive program, **T7.BAT**, the input file must have the **“.INP”** filename extension. An alternative would be to rename the previously generated ***.OUT** and ***.PLT** files to prevent overwriting. Similar precautions should be taken if operating outside of **T7.BAT** control.

4. Initiate execution by entering

C:\TMAP7>t7 <filename>

where <filename> is the root name (with no extensions) of the restart input file just generated. (here C:\TMAP7> is assumed to be the DOS prompt.)

6.0 CONCLUSION

The TMAP code has been very useful over the years since its first development. It has been used in numerous safety calculations, experiment simulations, and design checks. Changes made to the TMAP code to convert it to TMAP7 should improve the functional usefulness of the code for a variety of applications where previous versions have been used. The elements of code utilization remain essentially the same, but added features of multiple traps, radioactive decay, and surface only species, combined with some refinements in the way gas partial pressures are calculated in boundary enclosures have resulted in significant changes in the coding.

A companion report, J. A. Ambrosek and G. R. Longhurst, September 2004, *Verification and Validation of TMAP7*, Ref. 1, provides numerous examples and input files which may be used to build from in creating input files (also in *samples* folder) for the user's particular needs.

ACKNOWLEDGEMENT

The author is indebted to the U.S. Department of Energy, Office of Science for supporting the publication of this document under Idaho Operations Office Contract DOE-AC07-99ID13727. Thanks are also due to Brad J. Merrill and James L. Jones, the authors of the TMAP code upon which TMAP7 was based, to Phil Sharpe and James Ambrosek for their helpful suggestions, and to Theron Marshall for his thoughtful review of the manuscript. Appreciation is also expressed to the users of earlier TMAP versions, who have made many helpful suggestions. Finally, thanks to the Free Software Foundation, Boston, MA for their benevolent work to make G77 and other software available at no charge under public license.

REFERENCES

1. J. A. Ambrosek and G. R. Longhurst, September 2004, *Verification and Validation of TMAP7*, INEEL/EXT-04-01657., Idaho National Engineering and Environmental Laboratory, Idaho Falls, Idaho.
2. B. J. Merrill, J. L. Jones, and D. F. Holland, April 11, 1988, *TMAP/MOD1, Tritium Migration Analysis Program Code Description and User's Manual*, EGG-EP-7407.
3. G. R. Longhurst, D. F. Holland, J. L. Jones, and B. J. Merrill, June 12, 1992, *TMAP4 User's Manual*, EGG-FSP-10315.
4. G. R. Longhurst, S. L. Harms, E. S. Marwil, and B. G. Miller, July 8, 1992, *Verification and Validation of TMAP4*, EGG-FSP-10347.
5. S. M. Myers et al., 1989, "Ion Beam Studies of Hydrogen-Metal Interactions," *Journal of Nuclear Materials*, 165, pp. 9-64.
6. M. I. Baskes, "A Calculation of the Surface Recombination Rate Constant for Hydrogen Isotopes on Metals," *Journal of Nuclear Materials*, 92, 1980, p. 318.
7. F. J. Blatt, *Physics of Electronic Conduction in Solids*, McGraw-Hill, New York, 1968, pp. 202-207.

APPENDIX BATCH FILE LISTINGS

T7.BAT

```
echo off
del prepout
del codeout
del tape1
del tape7.for
del tmapinp
copy %1.inp tmapinp
tmapp7.exe
if not exist tape7.for goto noeqn
g77 -c tape7.for
g77 tmapc7.o tape7.o
goto execute
:noeqn
g77 tmapc7.o equ.o
:execute
a.exe
del a.exe
copy prepout+codeout %1.out
copy pltdata %1.plt
echo Task Complete
```

G.BAT

```
SET OLDPATH=%PATH%
PATH=c:\TMAP7\bin;%PATH%
SET LIBRARY_PATH=c:\TMAP7\lib
```

Notes

1. If TMAP7 is installed in a different directory than “c:\TMAP7”, the user will need to edit G.BAT, replacing “c:\TMAP7” with the path to the folder where the installation was performed (2 places).
2. This batch file must be run at the start of any TMAP7 session to provide g77 access to the required libraries.